Transparent Path Planning For Uncrewed Air Traffic Management

Zou, Y.

DOI
10.4233/uuid:7a39fde3-2129-49a0-a0ed-f1b93c9d6fa2

Publication date
2025

Document Version
Final published version

Citation (APA)
Zou, Y. (2025). *Transparent Path Planning For Uncrewed Air Traffic Management*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:7a39fde3-2129-49a0-a0ed-f1b93c9d6fa2

TRANSPARENT PATH PLANNING FOR UNCREWED AIR TRAFFIC MANAGEMENT

Yiyuan Zou

TRANSPARENT PATH PLANNING
FOR UNCREWED AIR TRAFFIC MANAGEMENT

TRANSPARENT PATH PLANNING
FOR UNCREWED AIR TRAFFIC MANAGEMENT

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus, Prof.dr.ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on
Wednesday 17th December 2025 at 15:00 o'clock

by

Yiyuan ZOU

Master of Engineering in Transportation Planning and Management,
Nanjing University of Aeronautics and Astronautics, China
born in Jiangsu, China

This dissertation has been approved by the promotors.

promotor: Prof.dr.ir. M. Mulder
promotor: Dr.ir. C. Borst

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof.dr.ir. M. Mulder,	Delft University of Technology, promotor
Dr.ir. C. Borst,	Delft University of Technology, promotor
Independent members:	
Prof.dr.ir. P.N.A.M. Visser,	Delft University of Technology
Dr. O.A. Sharpans'kykh,	Delft University of Technology
Dr. N. Yorke-Smith,	Delft University of Technology
Prof.dr. D. Delahaye,	Ecole Nationale de l'Aviation Civile, France
Prof.dr. J. Lundberg,	Linköping University, Sweden
Prof.dr.ir. R.C. Alderliesten,	Delft University of Technology, reserve member

Keywords: Algorithmic Transparency, Path Planning, Interface Design, Uncrewed Air Traffic Management, User Study

Printed by: Ipskamp Printing

ISBN 978-94-6518-176-9

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

SUMMARY

With the rapid advancement of technology, drones are being actively deployed across various domains. To manage their growing presence in the airspace, Uncrewed Air Traffic Management (UTM) has been proposed and is currently under development. Given the expected high volume of drone traffic, UTM will rely heavily on high levels of automation, as it is impractical to control each drone manually in the manner of traditional Air Traffic Control (ATC). However, this reliance on automation presents potential risks, particularly in airspace around airports where crewed aircraft are taking off and landing. Since completely reliable automation has not yet been achieved, anomalies or failures in UTM systems could increase the risk of collisions between drones and crewed aircraft. Therefore, UTM still warrants human supervision to ensure the safety of drone operations.

As automation becomes more advanced and complex, it also becomes increasingly difficult for humans to supervise, thereby hindering their trust and acceptance. Previous research suggests that some form of "seeing-into" transparency may be required to address this issue and support effective human supervision of automated systems. In this dissertation, "seeing-into" transparency is categorised into *operational transparency* and *engineering transparency*. Operational transparency offers (real-time) insights into the automation's states, actions, goals, and environmental impact, helping operational users maintain situation awareness and respond effectively to changing conditions. Engineering transparency, in contrast, discloses the inner workings of automation, enabling users to develop a deeper understanding of automation behaviour. This research adopts a bottom-up approach, beginning with engineering transparency and progressing towards operational transparency.

This dissertation focuses on achieving transparent path planning in UTM routing. To this end, a visual approach was first proposed to reveal the internal processes of path-planning algorithms, with a focus on graph- and sampling-based ones, as shown in **Chapter 2**. To demonstrate the effectiveness of the approach, a novel web-based pathfinding visualiser was developed that incorporates various classic and advanced path-planning algorithms, such as A*, Theta*, Anya, Polyanya, Rapidly-exploring Random Tree (RRT), RRT*, Informed RRT* and Batch Informed Tree (BIT*). To evaluate the impact of the proposed approach on algorithm runtime, extensive benchmark tests were performed on public datasets. Results show that extracting all search trees during the search process may significantly slow down the original algorithms. For large-scale, real-time operations, it is recommended to record only necessary data during the search and perform search tree extraction afterwards for visualisation.

To further investigate the effectiveness of algorithmic transparency, a user study was conducted to evaluate its impact on human understanding, as presented in **Chapter 3**. Considering that directly presenting the search process may overwhelm users, particularly in operational contexts, the path-planning transparency was structured into six distinct levels. Results indicate that as the transparency level increases, so does human understanding. However, the relationship between transparency and understanding is not a linear one. When the algorithm behaves contrary to human expectations and increased transparency fails to provide a clear explanation, users may become even more confused than without the additional transparency. For non-expert users unfamiliar with the algorithm, full transparency is often critical for meaningful understanding. The user study suggests that sampling-based algorithms may be easier to comprehend than graph-based algorithms. While the randomness inherent in sampling-based algorithms makes their behaviour difficult to predict, their overall rationale and underlying principles are meaningful and intuitive to humans.

As the ultimate goal of this dissertation is to achieve transparent path planning for UTM, the focus was then shifted from path-planning algorithms to UTM routing, broadening the concept of algorithmic transparency from purely engineering concerns to encompass operational dimensions, as shown in **Chapter 4**. A unified transparency taxonomy was developed, integrating diverse aspects of algorithmic transparency. Based on the proposed taxonomy, twenty transparency elements and their corresponding visual prototypes were devised for UTM routing. A survey study was then conducted to investigate the needs and preferences of Air Traffic Controllers (ATCos) and drone experts regarding these elements and prototypes. Results show that operational transparency is deemed more useful than engineering transparency in nominal UTM scenarios, whereas engineering transparency becomes more valuable when UTM routing fails. In the survey, operators were also asked to group the transparency elements, and their groupings aligned with the proposed transparency taxonomy.

The survey study captures only the initial opinions of operators, shaped by their prior knowledge and experience. To gain more insights, a human-in-the-loop experiment was performed to further examine the actual usage of various transparency elements in dynamic scenarios, where time pressure is a key concern, as shown in **Chapter 5**. Results show that information regarding the Closest Point of Approach (CPA) between drones and crewed aircraft is the most useful element for supporting tactical UTM supervision. When UTM routing fails, operators typically seek more information, such as constraint changes and details about the algorithm's inner workings, to understand the failure and to gather clues that inform their intervention strategies. Similar to the user study presented in Chapter 3, the experiment in Chapter 5 also suggests that in UTM contexts, sampling-based algorithms might be more suitable for supervision than graph-based algorithms. This is likely because the search tree visualisation of sampling-based algorithms could more clearly convey the algorithms' exploration efforts, offering useful cues for human intervention, such as indicating regions that are likely to be conflict-free.

In conclusion, this research achieves algorithmic transparency in path planning and demonstrates its application within UTM contexts. It contributes further empirical evidence to the growing body of research underscoring the importance and benefits of algorithmic transparency. The findings suggest that algorithmic transparency can enhance human understanding, but its utility in operational settings may be limited by situations, time pressure, and workload. As operators develop trust or expertise, their need for transparency may diminish. **Overall, transparency is essential to facilitating trustworthy automation, especially when it is not yet fully reliable.**

CONTENTS

1

INTRODUCTION

This chapter first introduces the background and operational concept of Uncrewed Air Traffic Management (UTM), describes the research problem with a focus on transparency issues in UTM, and then presents the approach along with four key questions. The research scope is defined by clarifying the underlying assumptions and areas of focus. The dissertation outline is also included to illustrate how each chapter contributes to and aligns with the overall study.

1.1. BACKGROUND

With the rapid advancement of technology, Uncrewed Aerial Vehicles (UAV), commonly referred to as Uncrewed Aircraft Systems (UAS) or drones, have emerged as transformative tools across a wide range of commercial and industrial sectors in recent years. They are increasingly being utilised for applications such as aerial surveying, infrastructure inspection, agriculture, logistics, and emergency response. In China, the number of registered drones rose by 42.6% between 2018 and 2021, reaching 832,000 by the end of 2021 [1]. In the United States, over 842,000 small commercial drones (weighing between 250 grams and 25 kg) had been registered by 2023 [2]. In Europe, government and commercial units are expecting drone operations to increase from several thousand to hundreds of thousands from 2016 to 2035 [3]. This upward trend highlights the growing influence of drones in our daily lives.

However, as the number of drones increases, the risk they pose to humans also rises. A malfunctioning drone that falls from the sky may strike people on the ground, while an unauthorised and uncoordinated drone may interfere with crewed aviation, endangering passengers on board. In recent years, unannounced private drone activity near airports, such as Gatwick, Heathrow, Frankfurt and Madrid, caused significant disruptions to regular air traffic and even led to temporary airport shutdowns [4]. These risks and incidents underscore the need for regulation and management of drone operations.

From 2014 to 2016, the National Aeronautics and Space Administration (NASA) developed an initial Concept of Operations (ConOps) for UAS Traffic Management (UTM) in the United States [5, 6]. Using a risk-based approach, four Technical Capability Levels (TCLs) were defined to progressively advance UTM from limited drone operations in rural areas to high-density operations in urban environments. During the same period, the Single European Sky ATM (Air Traffic Management) Research (SESAR) Joint Undertaking also developed a ConOps for the European UTM, known as U-space, and released a corresponding blueprint [7]. Four levels of U-space services were proposed according to levels of drone automation and connectivity. Nowadays, the American UTM ConOps has been updated to its second version by NASA and the Federal Aviation Administration (FAA) [8], whereas the U-space ConOps from SESAR is now in its fourth edition [9]. The fifth edition is currently under development through the SESAR project CORUS five [10].

In general, both American and European UTM systems are expected to rely on high levels of automation, providing a wide range of automated services to support drone operations. These services include operation planning, routing, weather and terrain avoidance, intent sharing, and strategic and tactical de-confliction. Additionally, both systems apply digital airspace management techniques, such as geofencing and geocaging [11], to restrict drone flight positions. This approach treats airspace as a shared resource and dynamically allocates it among a large number of drones (and other airspace users), thereby reducing operational interference and potential conflicts between them.

However, the American and European UTM systems differ in their approach to supporting drone operations in controlled airspace. In the American UTM system, UTM services provide information and advisories, such as conflict detection and resolution, in a supportive manner [8, pp. 19–20, 36–39]. Drone pilots or operators maintain primary responsibility for ensuring operational safety. In contrast, the European UTM system in controlled airspace resembles traditional Air Traffic Control (ATC), offering tactical conflict resolution services [9, pp. 34-36]. Drone pilots or operators must follow instructions issued by UTM in controlled airspace. In existing controlled airspace for crewed aviation, ATC is expected to manage both drones and crewed aircraft with the assistance of UTM services. However, it remains unclear how to support ATC in achieving this goal. This research is guided by the European UTM (U-space) ConOps and seeks to address this gap, with a particular focus on ATM–UTM integration and human-automation interaction.

1.2. ATM–UTM INTEGRATION

Drone operations in (existing) controlled airspace, unlike those in uncontrolled airspace, often require coordination with crewed aviation. Nowadays, to access controlled airspace in the Netherlands, a drone operator needs to request permission from Luchtverkeersleiding Nederland (LVNL) and submit flight plans via GoDrone [12]. Figure 1.1 presents the GoDrone map in the vicinity of Rotterdam The Hague Airport. The red areas are mostly no-fly zones for drones, and the large red circle with an extended rectangle represents the Control Zone or Controlled Traffic Region (CTR), centred around the airport. Currently, tower controllers are responsible for monitoring both drones and crewed aircraft in the CTR and have the authority to approve or deny clearance for drone flights. Due to the availability and workload of tower controllers, the number of drone flights that take place simultaneously within the CTR is limited.

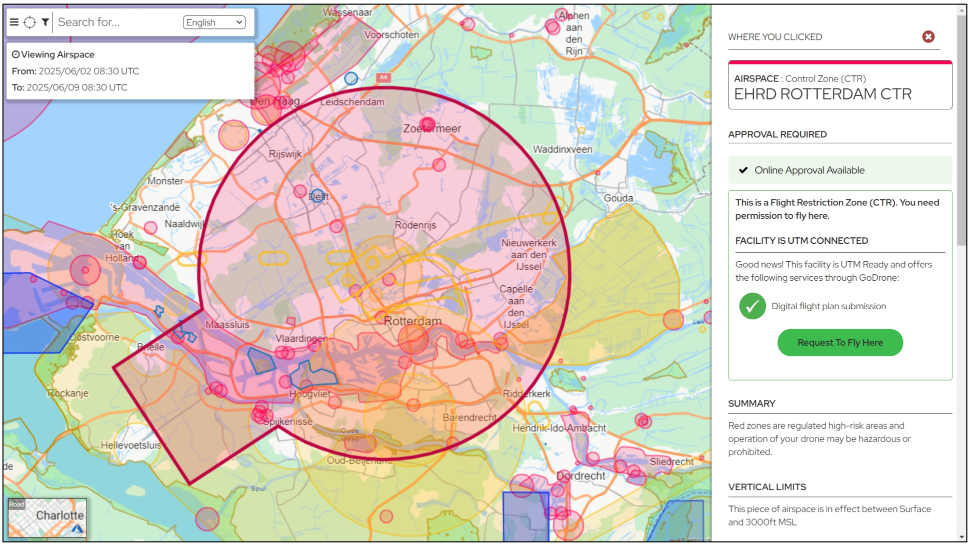


Figure 1.1: Screenshot of the GoDrone map in the vicinity of Rotterdam The Hague Airport [12]. The right side shows relevant information about the selected airspace, including its name, type, and altitude. The green button allows users to submit a digital flight plan to request authorisation from LVNL.

A CTR such as the one above Rotterdam The Hague Airport typically occupies a large portion of urban airspace. As the demand for drone operations within the CTR increases, current tower control capabilities may be insufficient to meet it. To facilitate the integration of drones into controlled airspace, U-space has advanced the development of collaborative interfaces with ATM [13–16]. The concept of Dynamic Airspace Reconfiguration (DAR) has been developed in the AURA (ATM U-space Interface) project [17]. In this concept, controlled airspace is defined as either ATM-controlled zones for crewed aviation or UTM volumes for drone operations. Figure 1.2 depicts two possible operational scenarios based on DAR. Scenario 1 indicates that controlled airspace is assigned to ATM by default, with ATM having the authority to delegate portions of the airspace to UTM when drones request access. Scenario 2 adopts the opposite approach, with UTM managing controlled airspace and delegating portions to ATM as required. Essentially, regardless of the scenario, the goal of DAR is to reduce potential interactions between drones and crewed aircraft by clearly defining the airspace boundaries.

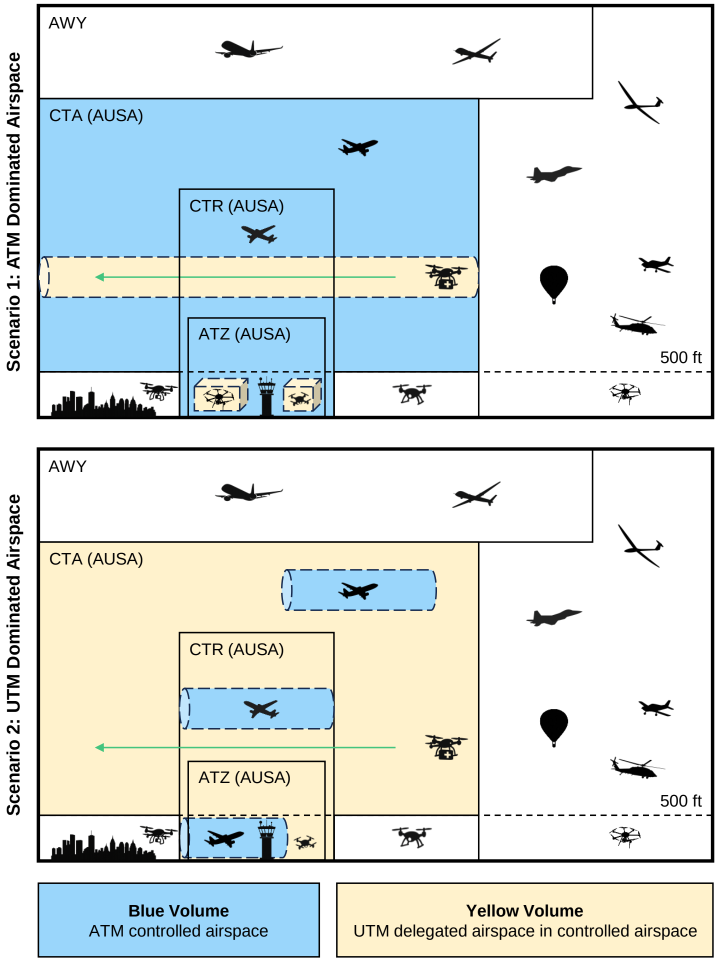


Figure 1.2: ATM–UTM integration in controlled airspace, adapted from the AURA project [17]. AUSA is the ATM–UTM Shared Airspace. ATZ is the Aerodrome Traffic Zone, which is part of the CTR. CTA is the Control Area, representing a broader portion of controlled airspace. This research primarily focuses on CTR, in alignment with the LVNL requirements.

To further support safe ATM–UTM integration, two key concepts have emerged based on DAR: segregation and separation [15]. Segregation refers to the complete spatial isolation of drones from crewed aircraft. In this concept, once controlled airspace is allocated based on flight plans and operational constraints, there is no interaction between ATM and UTM. Drones operate exclusively within UTM airspace, while crewed aircraft remain confined to ATM airspace. The likelihood of conflict between drones and crewed aircraft is minimised. However, full segregation may not always be feasible in practice. For example, emergency helicopters may sometimes need to enter UTM airspace to save time, and medical drones might have to traverse ATM-controlled areas near runways. Separation, in contrast, involves keeping beyond a predefined safe distance between drones and crewed aircraft. It offers greater adaptability in dynamic environments and can be achieved not only through DAR, but also via direct commands issued to drones, or, when necessary, to crewed aircraft.

Given the likelihood for drones to vastly outnumber crewed aircraft in the future, controlled airspace may eventually be dominated by UTM, as shown in Scenario 2 of Figure 1.2. Accordingly, this research focuses on Scenario 2, exploring how safe ATM–UTM integration can be achieved and how humans are involved in this context.

1.3. ROLE OF HUMANS IN UTM

In traditional ATM, pilots and Air Traffic Controllers (ATCos) both play a vital role in ensuring safe and efficient crewed aviation operations. Similarly, in UTM, drone pilots (or operators) and automated UTM services are crucial to enabling the safe and orderly integration of drones into the airspace. Drone pilots can be either human operators controlling drones manually, or autopilot systems enabling autonomous operations. This study considers only autonomous drones, disregarding response delays caused by human pilots. This assumption aligns with the U-space ConOps [9], where the level of UTM corresponds to the degree of drone automation, and full ATM–UTM integration is generally associated with the highest UTM levels (U3-U4).

In U-space, automated UTM services resemble ATC within controlled airspace, where issued instructions are mandatory. This indicates that the automated UTM system bears the ultimate responsibility for ensuring safety, and all drones should be capable of being directly controlled by it. However, this may be problematic in some cases simply because 100% safe and reliable automation does not exist yet. Automation failures in UTM could increase the risk of collisions between drones and crewed aircraft, posing a threat to human lives. Therefore, human oversight still remains essential to ensure the safety of drone operations, particularly in emergency or abnormal situations.

Initial studies explored whether UTM supervision could be integrated into the existing duties of ATCos alongside their regular ATC tasks [15], according to the suggestions in the U-space ConOps [9]. The findings, however, suggest that a dedicated supervisory role may be necessary. In this dissertation, the individual responsible for monitoring drone traffic and overseeing UTM operations is referred to as “UTM Operator/Supervisor”. The UTM supervisor does not need to be a professional ATCo, but can be someone with prior experience in drone operations, such as a drone operator or expert. Certainly, adequate training is required before they can assume the role of a UTM supervisor.

1.4. PROBLEM DEFINITION

To support human operators in supervising UTM operations, some form of "seeing-into" transparency is required that provides insights into the inner workings of UTM, enabling operators to understand its behaviour and limitations [18–20]. For instance, in a previous human-in-the-loop experiment regarding tactical UTM operations [13], some operators questioned why path A was chosen by automation for drone rerouting instead of their expected path B, expressing a desire for higher transparency to understand the underlying rationale of the path-planning algorithm. "Seeing-into" transparency aims to "*facilitate human-automation interaction by revealing the automation's responsibilities, capabilities, goals, activities, inner workings, performance, or effects to the human in real time*" [20, p. 794]. It contrasts with "seeing-through" transparency, which refers to a seamless interface that creates direct interaction between humans and automated tasks, making the interface itself appear invisible. In this dissertation, the term transparency only refers to "seeing-into" transparency.

Several transparency models that specify what information should be disclosed have been developed, including Lyons' human-robot transparency model [21] and the situation awareness-based agent transparency model [22]. These models have been successfully applied in various fields such as multi-mission uncrewed systems [23–25]. Algorithmic transparency, with a focus on algorithms and their inner workings, has increasingly attracted public attention as a means to understand and justify algorithmic systems [26–29]. It is defined as the "*disclosure of information about algorithms to enable monitoring, checking, criticism, or intervention by interested parties*" [26, p. 811]. The notion of progressive disclosure has been introduced to enable the incremental and on-demand provision of relevant algorithmic information [30].

In addition to "seeing-into" transparency, Explainable AI (XAI) has recently emerged as a field dedicated to explaining how AI systems function to specific audiences [31–33]. Instead of simply disclosing information, XAI seeks to generate human-understandable explanations that are usually more concise and meaningful for users. Transparency provides access, like a book open to all, whereas XAI serves as a teacher who interprets and tailors the content to learners. If explanations are regarded as a form of information, XAI can also be seen as a way to enhance transparency [34]. The "Right to Explanation" has been established under the European Union (EU) General Data Protection Regulation (GDPR) to protect individuals affected by algorithmic decisions [35].

Although some progress has been made in transparency, it remains unclear how existing models and frameworks can be integrated and applied to specific contexts such as UTM. Several SESAR projects have explored transparency in ATM [36], but different design approaches were employed. The goal of this research is to unify these efforts by proposing a comprehensive transparency framework and apply it in the context of UTM. Considering that automated UTM should be an algorithmic system (e.g., path-planning algorithms), the key problem this dissertation addresses is therefore:

Research Problem

How can algorithmic transparency be achieved in UTM to support tactical UTM operations within controlled airspace, ensuring that human operators can effectively understand and supervise automated UTM decision-making?

The study focuses on conflicts between drones and crewed aircraft. Collision avoidance among drones is assumed to be handled by onboard "sense and avoid" systems. It is also assumed that only drones can be controlled by UTM, and crewed aircraft are regarded as dynamic obstacles. More assumptions will be detailed in Section 1.6.

1.5. RESEARCH APPROACH

To address the research problem, the primary service that UTM will provide during tactical operations is identified first. Similar to ATC, the goal of UTM is to guide drones to their destinations while avoiding conflicts with other aircraft. This is essentially a Multi-Agent Path Finding (MAPF) problem [37]. Given the widespread use of the First-Come-First-Served (FCFS) strategy in aviation, MAPF is usually decomposed into conflict-free single-agent path planning, where other aircraft are treated as dynamic obstacles relative to the aircraft currently being planned [38]. Conflict-free path planning can generate routes that avoid both static and dynamic obstacles. It can be viewed as an extension of traditional path planning, enhanced with conflict detection and resolution capabilities.

Therefore, path planning forms a core capability of UTM, offering routing services for drones. The objective of this research can thus be reframed as achieving transparent path planning for UTM. In this dissertation, only graph-based and sampling-based path-planning algorithms are considered as promising candidates for UTM routing, excluding machine learning-based methods. This choice is motivated by the nature of the operational UTM environment, which is typically well-defined and fully observable via the UTM interface [14], and by the predictability of crewed aircraft trajectories. Moreover, graph-based and sampling-based algorithms benefit from solid mathematical foundations and theoretical guarantees, making them more reliable for real-world deployment. In contrast, machine learning-based methods still face significant challenges regarding their trustworthiness and certification, especially in aviation contexts [39]. It is foreseeable that, in the short term, graph-based or sampling-based path planning will be a more feasible and practical solution for UTM than learning-based path planning.

This dissertation distinguishes between two types of "seeing-into" transparency: *engineering transparency* and *operational transparency*. Engineering transparency reveals the internal mechanisms of automation, allowing users to build a deep understanding of how it functions. Operational transparency, on the other hand, provides (real-time) information regarding the automation's states, actions, goals, and environmental interactions/impact, supporting users in maintaining situation awareness. The research follows a bottom-up approach, starting with engineering transparency and advancing towards operational transparency, which together form the two main parts of the dissertation.

1.5.1. ENGINEERING TRANSPARENCY

Given that path-planning algorithms typically follow fixed procedures, the initial focus is on uncovering their internal processes without involving specific UTM scenarios. Due to the benefits of visualisation in presenting procedural knowledge [40–42] and various successful practices in pathfinding visualisers [43–48], this research also focuses on a visual approach rather than a textual or verbal approach for path-planning transparency. Most existing visualisers include only a limited set of classic algorithms, such as Dijkstra's [49], A* [50] and Rapidly-exploring Random Tree (RRT) [51], and lack a clearly defined, unified visualisation framework. Additionally, since visualisation is essentially a form of information presentation, extracting information from the algorithms' internal processes for visualisation can slow down the algorithms and thus affect the system's real-time performance. These issues lead us to the first research question:

Research Question 1

How can the internal processes of path-planning algorithms be visualised via a unified approach, and what impact does such visualisation have on algorithm runtime?

To address this research question RQ1, the common steps of various graph-based and sampling-based path-planning algorithms are firstly identified based on a literature review and practical experience. The experience is gained via the implementation of various classic and advanced path-planning algorithms. The common steps establish a foundation for path planning, upon which a general approach is proposed for storing, extracting and visualising information from the algorithm's internal process. To demonstrate the proposed approach, a novel pathfinding visualiser is developed, incorporating over ten representative path-planning algorithms. Benchmark tests are then conducted to assess the influence of the information extraction on the algorithm runtime based on public datasets [52].

The first step of the dissertation (RQ1) seeks to uncover and present detailed insights into the internal processes of path-planning algorithms from a technology-centred perspective. The relevant information disclosed is referred to as *engineering transparency*. The second step (RQ2) will then shift to a human-centred approach, exploring how algorithmic transparency affects human understanding. From the human-centred perspective, presenting too much information at once could overwhelm human users, and thus transparency should be structured and revealed according to user needs. To assess whether increased transparency truly leads to enhanced understanding, empirical user studies are required. Therefore, the second research question is formulated as follows:

Research Question 2

How can visual algorithmic transparency information in path planning be organised in a structured way, and how does that affect human understanding?

To address this research question RQ2, path-planning transparency is firstly chunked into distinct, meaningful information elements. Then, inspired by the notion of progressive disclosure [30] and previous Ecological Interface Design (EID) practices in aviation

[53, 54], these elements are organised into hierarchical levels that follow a typical human top-down, problem-solving strategy. As the transparency level increases, deeper information is progressively disclosed.

Based on the varying levels of transparency, an experiment is designed to assess how human understanding evolves as transparency increases. The experiment resembles an open-book exam, where the “book” is the transparency itself. As users gain more information, they are expected to develop a deeper understanding. If understanding improves noticeably with transparency, this would suggest that transparency is effective. As previously mentioned, this research focuses on graph- and sampling-based path-planning algorithms. Therefore, one algorithm from each category is selected in this experiment. This choice not only strengthens the reliability of the findings but also enabling an exploration of whether algorithm type influences human understanding.

1.5.2. OPERATIONAL TRANSPARENCY

The first part of the dissertation (RQ1 and RQ2) explores how to achieve transparent path planning and validate the effectiveness of algorithmic transparency in enhancing human understanding. These explorations are conducted without a specific operational context. Building on this foundation, the next phase of the research broadens the scope to examine the role of transparency in tactical UTM operations. In this case, algorithmic transparency extends beyond engineering transparency (the inner workings of algorithms) to include *operational transparency* that supports operator situation awareness [55]. This leads to the third research question:

Research Question 3

What constitutes algorithmic transparency in tactical UTM operations, and what are operators’ perceptions regarding its role in supporting UTM supervision?

To address this research question RQ3, a comprehensive literature review spanning various relevant fields is conducted, such as XAI [33], agent transparency [19], EID [56] and Cognitive Work Analysis (CWA) [57, 58], transparent ATM and UTM [36, 59]. In summary, there is no unified method to guide transparency design, especially in operational contexts. Existing studies typically focus on diverse aspects of algorithmic transparency. For example, the SESAR projects ARTIMATION [60], MAHALO [61] and TAPAS [55] all address similar problems in ATM but with different design choices [62].

Therefore, a unified taxonomy for algorithmic transparency is developed by integrating the concepts of operational and engineering transparency, with each category representing a distinct aspect of transparency. Operational transparency reveals an algorithm’s solutions, purposes/intents, and its impact on the broader context or environment to enhance operator situation awareness. In contrast, engineering transparency emphasises disclosing an algorithm’s internal mechanisms to improve operator understanding and foster their trust and acceptance. Based on the proposed taxonomy, various transparency elements and visual prototypes are designed for tactical UTM operations. A survey study is then conducted to investigate operators’ perceptions regarding the usefulness of these elements and prototypes in UTM supervision.

The third step of the dissertation (RQ3) develops a range of transparency elements for UTM and investigates operators' perspectives through a survey study (static scenarios). Based on this, the final step (RQ4) delves deeper into how operators utilise these transparency elements to supervise real-time UTM operations through a human-in-the-loop experiment (dynamic scenarios). This could also indicate whether the transparency elements preferred by operators (RQ3) are actually used in practice, especially under time pressure. Therefore, the final research question can be formulated as follows:

Research Question 4

How do operators utilise algorithmic transparency in tactical UTM supervision?

To address this research question RQ4, a novel UTM interface is developed, incorporating the transparency elements designed in response to RQ3. Drawing on the feedback from the survey study (RQ3), the transparency elements are further refined in the interface. A human-in-the-loop experiment is conducted to explore the usage of the transparency elements across different scenarios in tactical UTM operations. In line with the experiment performed to answer RQ2, a graph-based algorithm and a sampling-based algorithm are deployed and compared to assess whether algorithm type affects transparency usage and to identify which algorithm may be better suited for UTM supervision.

1.6. RESEARCH SCOPE

To limit the research scope, the following considerations are taken:

UTM ConOps. As UTM is still under development at the moment of writing, there is no universally recognised standard for how UTM will operate in the future. This dissertation follows the U-space framework and focuses mainly on tactical UTM operations in CTR near airports. It is assumed that conflicts between drones are resolved by their onboard detect-and-avoid systems. The primary role of UTM is therefore to efficiently navigate drones to their destinations while avoiding conflicts with crewed aircraft. The conflict here is defined as a loss of horizontal separation; vertical separation is not considered [14]. This choice is made because drones are highly susceptible to wind, and updrafts can lead to a sudden reduction in vertical separation. According to the U-space ConOps, UTM in controlled airspace should resemble traditional ATC, and thus a *centralised* UTM concept is implemented in this research, as presented in Figure 1.3. Operators can supervise automated UTM operations via an interface and, if necessary, issue instructions or implement constraints (e.g., geofences) to intervene in UTM decisions.

UTM simulations. The movements of crewed aircraft and drones are simulated using kinematic models rather than precise dynamic models. A fixed bank angle of 20° is applied to all turning manoeuvres of fixed-wing aircraft. Multi-rotor drones are assumed to execute turns at waypoints with a rotation rate of 30°/s. The climb and descent rates of crewed aircraft are determined by the current altitude and the target altitude of the next waypoint, ensuring a smooth, gradual transition. All drones maintain a constant altitude.

Geofencing. The dissertation focuses on grid-based geofencing [14, 63] for DAR. All drones must avoid entering designated geofenced areas. Geofencing is used to separate drones, particularly in high-density operations, from crewed aircraft. Operators can ac-

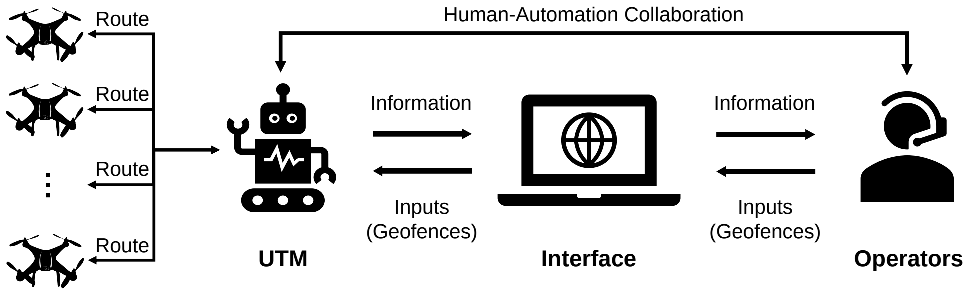


Figure 1.3: A centralised UTM system for drone (re-)routing.

tivate geofences to temporarily restrict drone access to specific regions. Geofences thus serve as a means for operators to intervene in the automated UTM routing service, which is governed by a certain path-planning algorithm. Geocaging, which keeps drones confined within a specific region, is not considered [11].

Path-planning algorithms. The dissertation focuses on graph-based and sampling-based path-planning algorithms, since they typically offer theoretical guarantees of finding feasible and (sub-)optimal paths [64, 65]. The explicit and fixed procedures of graph-based and sampling-based algorithms also make their internal processes relatively easy to disclose. Moreover, the intents and trajectories of all aircraft should be known to UTM, leading to a relatively stable environment with less uncertainty. This makes the routing problem in UTM particularly suitable to graph-based and sampling-based algorithms.

Transparency presentation. The dissertation focuses on visual transparency rather than verbal or textual transparency due to the unique benefits of visualisation in conveying time-related procedural knowledge [40–42] and many successful practices in path-planning visualisation. Additionally, the system implements *adaptable* transparency, allowing users to choose what information is shown, rather than *adaptive* transparency, which adjusts the display automatically based on user behaviour or context.

Infrastructure assumptions. It is assumed that all necessary infrastructure is in place to enable seamless communication between drones and UTM. All drones are considered autonomous and directly controllable by UTM. Communication delays and positioning uncertainties are not considered, and data transfer is assumed to be flawless. These assumptions do not affect the main conclusions of this dissertation, as the primary focus is on transparency issues in human-machine interactions.

1.7. DISSERTATION OUTLINE

This dissertation includes six chapters, with four core chapters corresponding to the four research questions, as shown in Figure 1.4.

Chapter 2: Transparent path planning (RQ1). Chapter 2 summarises the common steps of graph-based and sampling-based path-planning algorithms and proposes a general approach for storing, extracting and visualising relevant information from them. A novel web-based pathfinding visualiser is developed where many representative path-planning algorithms have been implemented. The impact of the proposed approach on

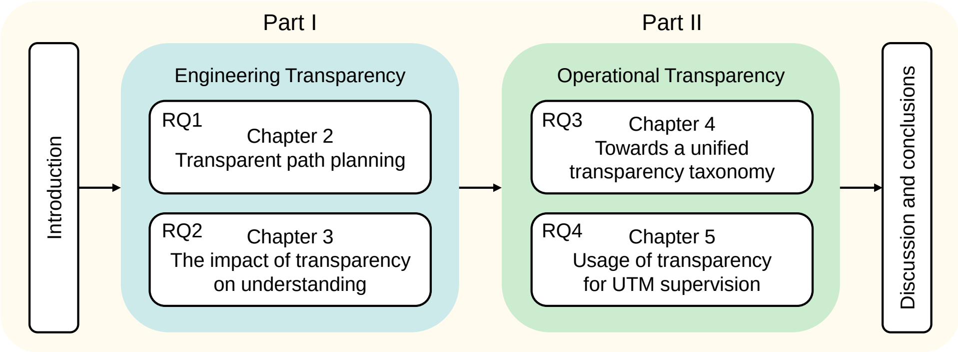


Figure 1.4: Dissertation outline.

algorithm runtime is also explored.

Chapter 3: The impact of transparency on understanding (RQ2). Chapter 3 further organises path-planning transparency into distinct levels and then explores their impact on non-experts' understanding of path-planning algorithms via a user study.

Chapter 4: Towards a unified transparency taxonomy (RQ3). Chapter 4 extends the research scope to tactical UTM scenarios, aiming to achieve transparent UTM. A unified taxonomy is developed for algorithmic transparency, integrating operational, domain, and engineering transparency. A survey-based user study is conducted to investigate the transparency needs of operators for UTM supervision and validate the effectiveness of the proposed taxonomy.

Chapter 5: Usage of transparency for UTM supervision (RQ4). Chapter 5 presents a human-in-the-loop experiment that investigates how operators use transparency information in real-time, tactical UTM operations. It also examines whether the transparency information operators preferred in the survey (Chapter 4) aligns with what they actually use and need in practice.

Chapter 6: Discussion and conclusions. Chapter 6 first revisits the research problem and summarises the main findings by answering the four research questions. Then, the potential directions for extending this research are discussed by relaxing some of the underlying assumptions. Finally, it concludes by distilling the main takeaways and highlighting the most significant findings.

2

TRANSPARENT PATH PLANNING

The goal of this chapter is to achieve transparent path planning by visually revealing the internal processes of path-planning algorithms. The common steps of graph-based and sampling-based path-planning algorithms are summarised, and a general approach is proposed for storing, extracting and visualising relevant information from them. A novel web-based pathfinding visualiser is developed where many representative path-planning algorithms have been implemented. The impact of the proposed approach on algorithm runtime is investigated.

The contents of this chapter are based on:

Paper title	Algorithmic transparency in path planning: A visual approach to enhancing human understanding
Authors	Yiyuan Zou and Clark Borst
Published in	International Journal of Human-Computer Studies **203**
DOI	10.1016/j.ijhcs.2025.103573

ABSTRACT

In step with technology evolution, various innovative algorithms have been proposed to solve path-planning problems. However, as algorithms become more complex, it also becomes increasingly more difficult for humans to understand them, hindering their applications. One way to overcome the barriers to understanding and thus help promote certain path-planning algorithms is to implement some form of 'seeing-into' transparency. This chapter presents a visual approach to achieve transparent path planning, aiming to visually reveal the internal processes of various classic and advanced path-planning algorithms. This approach is centred around storing, extracting and visualising information directly from algorithms themselves rather than developing additional explainers to generate post-hoc explanations for their outputs. A novel web-based interactive pathfinding visualiser was developed to demonstrate the effectiveness of the proposed approach. Over ten representative path-planning algorithms have been implemented, such as A, Theta*, Anya, Polyanya, Informed Rapidly-exploring Random Tree* (Informed RRT*) and Batch Informed Tree (BIT*). Benchmark tests were conducted to assess the additional computational time required by the proposed approach. It is hypothesised that visualisations could help in understanding the commonalities and differences of various path-planning algorithms, allowing one to: 1) identify the strengths and weaknesses of algorithms, 2) visually debug incorrect implementations, 3) seek opportunities for improving algorithms and 4) better match algorithms with suitable application domains.*

2.1. INTRODUCTION

Path planning is one of the fundamental problems in many domains, aiming to find the optimal path between two locations or states under certain environmental constraints. A large number of algorithms have been proposed to solve various practical path-planning problems. For example, in tackling the Euclidean shortest path problem, a series of advancements has been made, including A* [50], Theta* [66], Anya [65], 2^k-neighbourhoods [67], RayScan [68] and End Point Search [69]. However, as algorithms grow more complex, they become increasingly difficult for humans to understand and apply. For non-expert users, this lack of understanding may hinder their acceptance and trust in algorithms, further limiting the algorithms' real-world applications [70].

To address this issue, explainable path planning has been developed in recent years [71, 72]. It is a subfield of Explainable AI (XAI) [31, 33] and Explainable AI Planning (XAIP) [73, 74]. Explainable path planning aims to make path-planning algorithms more easily understandable to humans by explaining why certain paths were generated by the algorithms. For instance, it can clarify why path A is optimal rather than path B [71, 75], why path A is safe and feasible [72] and why the algorithm failed to find a path [76, 77]. However, this method does not truly explain how the algorithm works. These explanations emphasise the *output* more than the *internal process* that leads to a specific output.

Therefore, transparent path planning is introduced as a complement to existing explainable path planning. Rather than offering specific explanations for the output, transparent path planning focuses on disclosing the internal process, aiming to make relevant information regarding path-planning algorithms accessible to humans. Gaining insights into the internal process may also enhance understanding of the output, as the process is governed by the algorithm's underlying rationale, procedure and constraints, with the output serving as its final step or result.

Visualisation, particularly animation, could be an effective way to achieve transparent path planning due to its unique advantages in presenting time-related procedural knowledge [40–42]. Many pathfinding visualisers have thus been developed [43–48, 78, 79]. However, most visualisers either serve as one-off demonstrators [43, 78] or are confined to specific domains [44, 46]. There is no clearly defined and unified method among them. Considering the fact that numerous graph-based and sampling-based path-planning algorithms find paths by constructing search trees [51, 64–66, 80, 81], a general approach is proposed to extract search trees from the algorithm's internal process for visualisation. To show the effectiveness of this approach, a novel web-based pathfinding visualiser was developed, incorporating more than ten representative path-planning algorithms, such as A* [50], Theta* [66], Anya [65], Polyanya [82], Rapidly-exploring Random Tree (RRT) [51], RRT* [64], Informed RRT* [83], Batch Informed Trees (BIT*) [81], Time-Optimal Any-Angle Safe-Interval Path Planning (TO-AA-SIPP) [84] and Zeta*-SIPP [85]. Benchmark tests were conducted to assess the impact of search tree extraction on algorithm runtime.

This chapter is structured as follows: Section 2.2 summarises the common steps of graph- and sampling-based path-planning algorithms. Section 2.3 clarifies the concepts of explainability and algorithmic transparency and reviews relevant literature on explainable motion/path planning and visualisations. Section 2.4 presents a general approach for storing, extracting and visualising search trees from path-planning algorithms and analyses the impact of search tree extraction on algorithm runtime through benchmark tests. Section 2.5 discusses the potential benefits, applicability and limitations of transparent path planning.

2.2. PATH-PLANNING ALGORITHMS

In this section, the general path-planning problem and its relevant terminology are first clarified. Then, various graph-based and sampling-based path-planning algorithms are reviewed to identify and summarise their common steps. This section provides the foundation for achieving algorithmic transparency in path planning.

2.2.1. PATH-PLANNING PROBLEM

Path planning is a problem of determining a path from a start point p_s to a target point p_t given a search space S: (p_s, p_t, S). To simplify the problem, this chapter only focuses on 2D path planning with static obstacles where $S \subset \mathbb{R}^2$. The static obstacles are defined as $O = O_1 \cup O_2 \cup ... \cup O_n \subset \mathbb{R}^2$. Therefore, the free space in the environment without static obstacles is represented by $S_{free} = S \setminus O$. Then, the 2D path-planning problem can be rewritten as (p_s, p_t, S_{free}). Please note that S_{free} here is continuous, which is suitable for both graph- and sampling-based path-planning algorithms.

A *path* is a sequence of points $\pi = \{p_1, p_2, ..., p_n\} \subset S_{free}$ where $p_1 = p_s$ and $p_n = p_t$. The *cost* of a path π is the sum of the costs between each successive pair of points: $C(\pi) = \sum_{i=1}^{n-1} c(i, i+1)$ where $c(i, i+1)$ is the cost between the points p_i and p_{i+1}. The cost is usually represented by the Euclidean distance between p_i and p_{i+1}, and in this case, the general path-planning problem can be reduced to the Euclidean Shortest Path Problem

[69]. A path is *optimal* (shortest) if its cost is the minimum among all paths between its start and target points.

Many terms are interchangeably used in the field of path planning. For a better illustration, the terminology is clarified first in this dissertation, as shown in Figure 2.1. The green points and lines represent graphs (vertices and edges), while the orange points and lines refer to search trees (nodes and branches). Line-of-sight checks are utilised to determine if two non-adjacent vertices are mutually visible, which can be considered as a method to introduce more edges in a graph or more branches in a search tree.

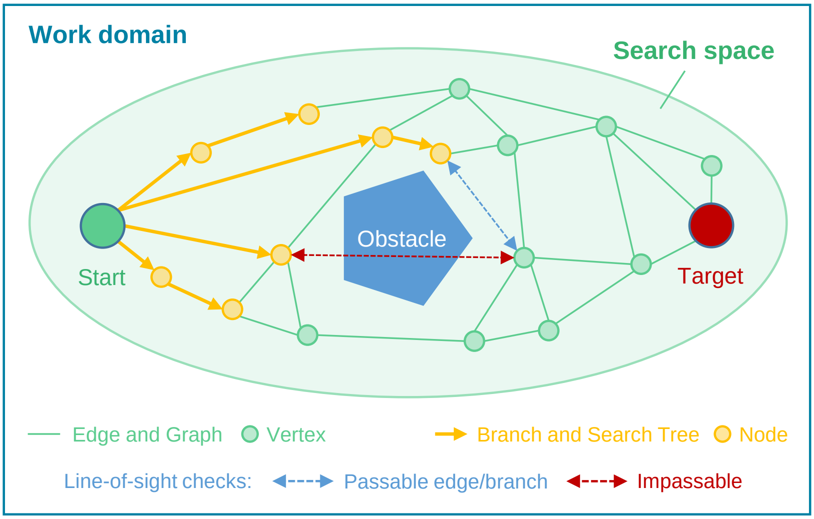


Figure 2.1: Terminology used in path planning.

In path planning, $g(n)$ represents the real cost from the start node n_s to the node n and is initially set to infinity. If $g(n)$ is less than infinity, it indicates that a feasible path to n has been found. $g(n,n')$ is the real cost from n to n'. $h(n)$ indicates the estimated cost from the node n to the target node n_t, which is usually assumed to be the cost of the direct path between n and n_t. $h(n,n')$ is the estimated cost from n to n'. Therefore, the total cost of a node is $f(n) = g(n) + h(n)$.

2.2.2. GRAPH-BASED PATH PLANNING

Since a continuous space contains infinite points, graphs are usually defined to limit the complexity of the space for path planning. Graph-based path planning firstly generates a graph $G = (V, E, C_E)$ where V is a set of vertices, $E \subseteq V \times V$ is a set of edges and $C_E : E \to \mathbb{R}^+$ is a cost associated with each edge. Then a search algorithm can be applied to search for the optimal solution within the limited search space. One benefit of such a graph is that the connections between vertices are predefined before searching, and thus it is easier to expand nodes to generate search trees. Without the predefined edges, the algorithm has to perform line-of-sight checks to expand nodes and build branches, which is generally more time-consuming for the search.

There are various approaches to generate graphs, such as regular grids, navigation meshes and visibility graphs [86]. Regular grids are perhaps the most common approach for discretising continuous spaces. Graph vertices on regular grids can be either the grid vertices or the grid centres. Different from regular grids, navigation meshes are collections of convex polygons that can more accurately represent the environment. Regular grids can be viewed as a special case of navigation meshes. Visibility graphs are graphs whose edges represent visibility connections between corresponding vertices. They are usually constructed from the vertices of obstacles.

A* is one of the most classic graph-based path-planning algorithms [50]. It has been applied to various fields and still forms a basis for many advanced algorithms [65, 68]. However, traditional A*-based algorithms are mainly implemented on regular grids to find grid-by-grid paths. The connections for each grid are constructed only between it and its eight adjacent grids. This kind of search is limited to 45-degree increments and thus the paths are not the truly shortest [66].

To address this issue, any-angle path planning is proposed, such as Theta* [66], Block A* [87] and Anya [65], which removes the 45-degree limitation to generate straighter and shorter paths. TO-AA-SIPP [84] and Zeta*-SIPP [85] are optimal any-angle path planning combined with Safe Interval Path Planning (SIPP) [88] for dynamic environments. In addition to any-angle path planning on grids, many mesh-based and visibility-graph-based algorithms are also designed to find true shortest paths [69]. For instance, Polyanya [82] is an extension of Anya, which is designed for navigation meshes instead of regular grids. RayScan [68] is an online path-planning algorithm which partially builds a visibility graph by using ray shooting to discover obstacles and then scanning along their edges to find ways around them.

In summary, despite the variety of graph-based path-planning algorithms, most of them are still based on the A* search. The algorithms mentioned above are all A*-based. Therefore, Algorithm 1 summarises the main procedure of A*-based path planning. Here, $open$ is a queue, sorted by the f-value in Line 5. N refers to the nodes explored by the algorithm, which can be used to extract the search tree at each step. More details on the usage of N are provided in the next section.

2.2.3. SAMPLING-BASED PATH PLANNING

Another way to search continuous spaces is through sampling. Rather than predefining a graph, sampling-based algorithms *randomly* pick points in the search space (or called sampling space) for discretisation. The sampling points can be regarded as the vertices of an *implicit* Random Geometric Graph (RGG) [89] whose edges are more flexible than fixed predefined graphs. Sampling-based path planning can be regarded as constructing an *implicit* RGG and an *explicit* search tree in the search space S [90]. There are two main approaches to build edges in RGGs: connecting each vertex to a specific number of its nearest neighbours [91] (k-nearest RGGs), or to all its neighbours within a certain distance [92] (r-distance RGGs).

When considering the usage of RGGs, two types of sampling-based path-planning algorithms emerge: Probabilistic Roadmaps (PRM) [80] and RRT [51]. In PRM-based algorithms, a *complete* PRM/RGG with edges is firstly built by sampling after which a search algorithm is applied to find the optimal path within this PRM/RGG. The main difference

Algorithm 1 A*-based path planning

Input: Search space S, start p_s, target p_t
Output: Optimal path π

Graph $G \leftarrow$ `discrete`(S)
Start node $n_s \leftarrow$ `createNode`(p_s, G)
Tree nodes $N \leftarrow \{n_s\}$, $open \leftarrow \{n_s\}$
while $open \neq \varnothing$ **do**
    $n_c \leftarrow \arg\min_{n \in open} f(n)$
    $open \leftarrow open \setminus \{n_c\}$
    if `isReach`(n_c, p_t) **then**
        return `pathTo`(n_c)
    end if
    $N_c \leftarrow$ `generateSuccessors`(n_c, G)
    $open \leftarrow open \cup (N_c \setminus N)$, $N \leftarrow N \cup N_c$
    Rewire the search tree and update the node costs
end while
return *null*

between PRM-based and graph-based path planning is the method of generating graphs. The former is based on sampling within a local region while the latter is a decomposition of the entire search space. BIT* [81, 90] can be regarded as a variant of PRM-based algorithms. For each *batch* in BIT*, an implicit RGG is built (or revised) and then a heuristic search can be applied to incrementally expand the search tree.

Different from PRM-based algorithms, RRT-based algorithms do not directly connect sampling points to build edges in RGGs, but instead utilise the sampling points to provide expansion *directions* for search trees. In each iteration, after sampling a random point in the search space, the RRT-based algorithm finds the nearest node in the search tree to this random point, and then extends a new branch from the nearest node to it. A parameter, incremental distance [51], limits the length of this new branch. If the incremental distance is large enough, the sampling points may directly become the nodes of the search tree after sampling and light-of-sight checks. In RRT* [64], by checking if the newly added node is a better parent for its neighbours within a certain distance, the search tree can be rewired to find asymptotically optimal paths.

In summary, PRM-based path planning also applies search algorithms after building RGGs, and the A* search used can also be described by Algorithm 1. Therefore, only the procedure of RRT-based path planning is summarised, as illustrated in Algorithm 2. The basic ideas of RRT* (line 9) and Informed RRT* (line 12) are also integrated.

2.2.4. COMMON STEPS IN PATH PLANNING

Based on the analysis above, although graph-based and sampling-based path planning appear to be different, both essentially find paths by building search trees. Their main difference is how they create a graph to discretise the search space. Graph-based path planning generally predefines a graph based on the decomposition of the search space, which is usually fixed but has an efficiently ordered nature. The complexity of the path-

Algorithm 2 RRT-based path planning

Input: Search space S, start p_s, target p_t, termination condition T
Output: Optimal/Near-optimal path π

Tree nodes $N \leftarrow \varnothing$ and paths $P \leftarrow \varnothing$
Initial condition $t \leftarrow$ `init()`
while $t \neq T$ **do**
    Random point $p_i \leftarrow$ `sample`(S)
    Nearest node $n_{\text{nearest}} \leftarrow$ `nearest`(N, p_i)
    Search node $n_i \leftarrow$ `steer`$(n_{\text{nearest}}, p_i)$
    if `collisionFree`$(n_{\text{nearest}}, n_i)$ **then**
        $N \leftarrow N \cup \{n_i\}$
        Rewire the search tree and update the node costs
        if `isReach`(n_i, p_t) **then**
            $P \leftarrow P \cup$ `pathTo`(n_i)
            $S \leftarrow$ `updateSearchSpace`(P, S)
        end if
    end if
    $t \leftarrow$ `update`(t)
end while
return `bestPath`(P)

planning problem within a graph is also limited. Sampling-based path planning creates an *implicit* RGG by sampling, which has a good scalability and can be continuously improved by adding more samples. After discretising the search space, a search algorithm, like A* and RRT (the node expansion strategy in RRT can also be regarded as a "search" algorithm), can be applied to generate a search tree and find a feasible or optimal path. In RRT-based algorithms, the RGG vertices are only used to provide expansion *directions* for the search trees.

Therefore, the common steps in graph- and sampling-based path planning can be summarised as follows:

1) *Define a search space to limit the search process.* The search space S is usually the entire map if there is no other physical constraint related to agent performance, such as vehicle endurance and manoeuvre limitations. For instance, in grid-based path planning, it is common to include barriers or walls around the map to constrain the search and prevent it from extending beyond the specified boundaries. In Informed RRT* [83], according to the best path found so far, an elliptical region is defined to narrow the search space.
2) *Generate a graph to discretise the search space.* As the search space is continuous and has an infinite number of points, it is difficult to search directly in the space. Both graph-based and sampling-based algorithms create a graph to discretise the search space. Then a search can be performed to find a path.
3) *Apply a search algorithm to find a path based on the graph.* The real output of a search is a search tree and the final path can be obtained by extraction. There are four main elements in a search algorithm: *search nodes, successors, cost function*

and *search procedure*. The successors indicate the connections of nodes and can be used to construct the branches of search trees. The cost function is designed to evaluate the cumulative costs of nodes, and the search procedure describes the main search steps. Please note that the search nodes are not limited to points and can also take other forms, like triangles [65].

The diversity of path-planning algorithms are reflected by the different approaches adopted in the common elements. For example, different methods to graph generation in graph-based path planning can lead to regular-grid-based, mesh-based or visibility-graph-based algorithms. How to define an efficient cost function to accurately reflect the goal of path planning is also an important element in real-world applications.

2.3. EXPLAINABILITY AND ALGORITHMIC TRANSPARENCY

Algorithmic transparency is the core concept of transparent path planning. It is related to explainability, as both aim to enhance the comprehensibility of models or algorithms. Explainability indicates the "*details and reasons a model gives to make its functioning clear or easy to understand given a certain audience*" [33, p. 85], whereas algorithmic transparency refers to the "*disclosure of information about algorithms to enable monitoring, checking, criticism, or intervention by interested parties*" [26, p. 811].

Explainability is associated with XAI and the notion of explanation [33]. As stated in [29], both explainability and algorithmic transparency address "black box" issues. The main difference is that explainability pertains to the "lack of clarity on the reasoning of AI", aiming to providing tailored explanations for why a particular output is generated, while algorithmic transparency concerns the "lack of clarity on the functioning of algorithms", focusing on simply disclosing all relevant information including inputs, outputs and internal processes [29]. Enhancing explainability can actually be seen as a way to boost transparency, as it enables the extraction of more human-understandable information from models, which can then be revealed to humans [30, 34].

There are three main strategies for enhancing explainability [31]. The first strategy involves creating inherently interpretable models, known as "white box" models, such as linear regression and decision trees. The second strategy employs post-hoc methods, utilising simplified interpretable models to approximate the functionality of the original complex models ("black box"), examples of which include Local Interpretable Model-agnostic Explanations (LIME) [93] and SHapley Additive exPlanations (SHAP) [94]. The third strategy focuses on developing models with partially explainable features. An example is the decomposed reward Deep Q-Networks (drDQN) that divides rewards into various meaningful components, thereby making the rationale behind actions understandable through the comparison of sub-rewards [95].

Some researchers have raised concerns that the explanations generated by post-hoc methods may not accurately reflect the functioning of original "black box" models [96]. They argue that such explanations should rather be termed "observations", because they merely approximate the input-output relationships by observing the black box's behaviour without truly explaining its mechanisms and rationale [97]. Therefore, in high-stakes domains, caution is advised when implementing post-hoc methods due to the potential risks arising from discrepancies between the original model and the post-hoc explainable model, and, if possible, "white box" models are preferred [96].

This research focuses on algorithmic transparency rather than explainability, as the main interest is in revealing the inner workings of path-planning algorithms instead of specifically explaining why a certain path is chosen. Since the final path is a direct outcome of the internal process, understanding the process should naturally lead to an understanding of the output. Additionally, the nature of graph- and sampling-based path-planning algorithms allows their internal processes to be relatively easily disclosed, similar to "white box" models in XAI. These factors motivated the choice of algorithmic transparency as a means to enhance human understanding in path planning.

2.3.1. EXPLAINABLE MOTION AND PATH PLANNING

While XAI primarily addresses black-box, learning-based approaches, XAIP represents a burgeoning field with a focus on elucidating automated planning and decision-making processes [73]. Explainable motion/path planning, a specific subfield of XAIP, strives to render the motion/path plans generated by AI more comprehensible to humans.

Brandao et al. [76] summarised different kinds and purposes of explanations within motion planning and developed optimisation-based and sampling-based explainable motion planners. Their attention was not limited to failure explanations (why did you fail?), but extended to contrastive explanations (why plan A rather than B?). Contrastive map-based explanations were introduced to clarify path plan optimality [71, 75]. These explanations were framed as solutions to inverse-shortest path problems, calculating the minimal modification required to the map to make the user's desired path optimal. This means if a user inquires why path A is optimal instead of path B, the explainable planner modifies the map to demonstrate scenarios where path B would be optimal. This approach has been extended to the failure-explanation problem in motion planning [77]. Furthermore, a visual approach based on temporal segmentation for multi-agent motion/path planning was proposed to explain plan feasibility [72, 98, 99]. This approach decomposes a plan into segments in which the agents' paths are disjoint, thereby illustrating that the plan is conflict-free and feasible.

In summary, the existing methods for explainable motion/path planning focus on explaining the optimality and feasibility of the final plan (output), often leveraging contrastive explanations. Referring to this, when disclosing the internal process of a path-planning algorithm, some form of cost-based contrastive explanations can also be integrated to illustrate why the algorithm follows a certain search direction and why an explored path is not the final path. For example, in an A*-based algorithm, the current node has the lowest cost value among the nodes in the $open$ list. From the cost values of the open nodes, the next current node (next search step) can be easily predicted.

2.3.2. PATH PLANNING VISUALISATION

The aforementioned approaches focus on elucidating the outputs of motion/path planning rather than the internal mechanisms behind them. Distinct from various ML models, the internal processes of motion/path planning, particularly in 2D environments, lend themselves more readily to visualisation. This accessibility has led to the development of numerous visualisers, providing a reference for transparent path planning.

Red Blob Games [78] and Moving AI Lab's Single Agent Search [43] were online tools tailored for educational purposes, featuring interactive demonstrators and accompanying learning resources. These tools provide detailed explanations of classic graph-based path-planning algorithms. PathFinding.js [44] and Mihailescu's visualiser [45] were primarily developed for graph-based path-planning, offering many adjustable options to modify the inputs and settings. Upon execution, an animation that illustrates the search process will be displayed. The Visualiser ThreeJS [79] attempts to portray several classic graph-based path-planning algorithms in 3D environments. While the algorithms are designed for 2D, the perspective can be smoothly transformed into 3D. This feature has the potential to be integrated into map navigation.

FAST Lab's Sampling-based Visualiser [46] and Local Planner Visualisation Project (LPVP) [47] focus on sampling-based path-planning algorithms rather than graph-based ones, presenting the exploration processes of search trees through animation. LPVP can also portray potential-field-based path-planning algorithms by presenting their vector fields. PathBench [48] is an integrated platform designed for portraying graph-based, sampling-based and learning-based path-planning algorithms, supporting both 2D and 3D environments. It includes an evaluation module for benchmarking different path-planning algorithms. Posthoc [100] is a debugging and visualisation platform for search algorithms. It takes as input a structured data log describing basic search operations and provides a visual interface used for playback, analysis and visualisation.

Given the goal of this research to achieve algorithmic transparency in path planning, the primary inspiration is drawn from path planning visualisations rather than explainable path planning methods. Instead of providing specific explanations for the output, the proposed approach aims to disclose all relevant information, allowing users to form their own judgments. By presenting how the algorithm works, users could understand how it navigates within the map, what constraints it considers, and why it generates a specific path. Due to the benefits of visualisation in presenting time-related procedural knowledge [40–42] and various successful practices in pathfinding visualisers, a visual approach is adopted rather than a textual or verbal approach for transparency. However, the existing visualisers do not systematically illustrate how to extract and present information from algorithms. The impact of information extraction on algorithm runtime also remains unexplored. This research aims to fill this gap.

2.4. VISUAL APPROACH TO TRANSPARENT PATH PLANNING

2.4.1. INFORMATION EXTRACTION

As the essence of visualisation is to present information, this section introduces a general approach for extracting information in path planning based on the common steps summarised in Section 2.2. The information extraction is performed during the search process. Depending on when the data are transmitted from the algorithm to the interface within the system, the visualisation can be either real-time (during the search) or post-hoc (after the search). The proposed approach supports both since the data transmission is independent of the data extraction. This research chooses post-hoc visualisation, which is displayed only when users request more transparency.

As stated in Section 2.2, search trees are at the heart of graph-based and sampling-based path-planning algorithms. The final path represents only one of many possible solutions explored by search trees. However, to save memory, the search trees are typically not stored directly during the execution of path-planning algorithms. Only the "latest" or "current" search tree is retained in a specific structured format. Path planning can be regarded as a process of continuously updating this "current" tree until it contains the optimal path. In practice, the tree data is usually embedded within the variables of tree nodes N (see Algorithms 1 and 2), requiring further extraction from them. To effectively extract search trees, two methods are introduced that specify the necessary data to be stored, along with procedures for extraction, as shown in Algorithms 3 and 4.

Algorithm 3 is perhaps the most straightforward method to extract search trees. The collection of tree nodes N is stored at each search step. By constructing branches from the tree nodes N, the search tree at a certain step can be extracted. The search process can then be portrayed by displaying the extracted search trees in order. However, storing all search trees using Algorithm 3 is not efficient. Much redundant data may be recorded since most branches are fixed after construction. For example, in A*-based algorithms, when a node n is inserted into the $closed$ list, the branch connecting $parent(n)$ and n is fixed and only needs to be extracted once, as depicted in Figure 2.2. The changes to the search tree are primarily caused by rewiring among the explored nodes or expansion to new nodes. It is only necessary to record the new changes or new branches at each step.

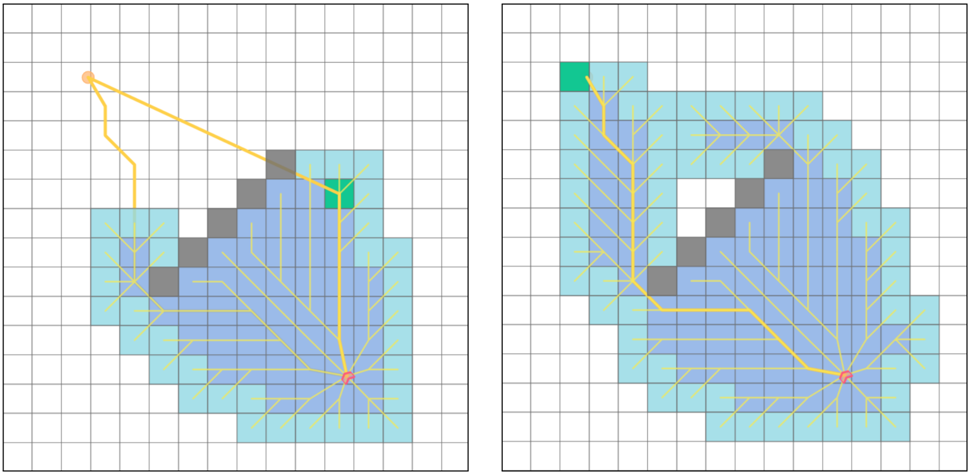


Figure 2.2: Part of the search tree is fixed during the search process. The light blue grids indicate the open nodes, whereas the dark blue grids denote the closed nodes. The yellow lines are the branches of search trees and the light orange thick line is the current optimal path found by the algorithm.

Let L_i denote the new branches at step i, and thus the cumulative collection of new branches at step i is $\Gamma_i \leftarrow \Gamma_{i-1} \cup L_i$. Based on Γ_i, the extraction process for search trees is slightly different. Algorithm 4 introduces forward and backward extraction methods based on Γ_i. Lines 17 and 19 ensure that the new branches will always be connected first and then the old useless branches will be discarded. The forward extraction method is particularly suitable for incremental data extraction as it extracts search trees according to the previous tree (T_{i-1}) and new branches (L_i). The backward method is more efficient for extracting only one search tree since it requires no pruning. Using Algorithm 4, only necessary data Γ_i need to be recorded during the search process. The benefit of Algo-

Algorithm 3 Extract i-th search tree directly

function `directExtract`(N_i)
  Search tree $T_i \leftarrow \varnothing$
  for each $n \in N_i$ **do**
    if $parent(n) \neq null$ **then**
      $T_i \leftarrow T_i \cup \{\, parent(n), n \,\}$
    end if
  end for
  return T_i
end function

Algorithm 4 Extract i-th search tree from branch sets

function `forwardExtract`(Γ_i)
  Search tree $T_i \leftarrow \varnothing$
  for $j = 1, 2, \cdots i$ **do**
    Branches $L_j \leftarrow \Gamma_i(j)$
    for each $\{\, parent(n), n \,\} \in L_j$ **do**
      $T_i \leftarrow T_i \setminus \{\, oldparent(n), n \,\}$
      $T_i \leftarrow T_i \cup \{\, parent(n), n \,\}$
    end for
  end for
  return T_i
end function

function `backwardExtract`(Γ_i)
  Search tree $T_i \leftarrow \varnothing$
  for $j = i, i-1, \cdots 1$ **do**
    Branches $L_j \leftarrow \Gamma_i(j)$
    for each $\{\, parent(n), n \,\} \in L_j$ **do**
      if $connected(n) = false$ **then**
        $T_i \leftarrow T_i \cup \{\, parent(n), n \,\}$
        $connected(n) \leftarrow true$
      end if
    end for
  end for
  return T_i
end function

rithm 4 is that it separates data storing from data extraction, meaning that it can obtain solutions first and then extract data later for visualisation. In summary, Algorithm 3 has to extract and store all search trees during the search process, which could significantly reduce the real-time performance of path-planning algorithms. In contrast, Algorithm 4 only needs to record new branches at each step, allowing the extraction of search trees to be performed after the search.

In practice, some other useful data can also be recorded for visualisation:

1) *Current node.* The current node is either the most promising node so far, or a new node added to the search tree.
2) *The path through the current node.* The current path is usually constructed by the explored path from the start to the current node and the expected path from the current node to the target.
3) In A*-based algorithms, the *status of a node* (e.g., whether open or closed) can be recorded and then presented with different colours in the visualisation [44].
4) In RRT-based algorithms, one can record (i) which node is the nearest tree node to the current node, (ii) the current sampling space, (iii) the current optimal path, (iv) and the rewiring radius.

2.4.2. INFORMATION VISUALISATION

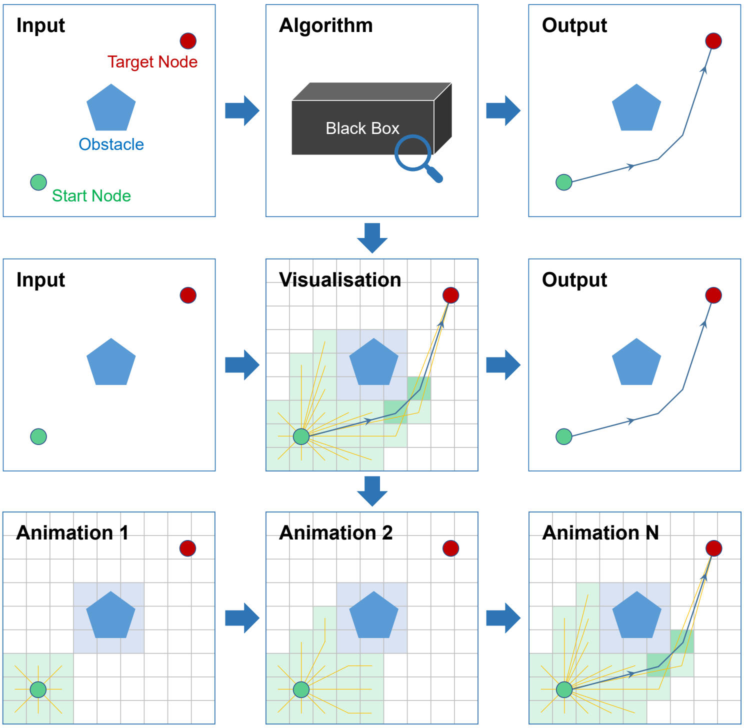
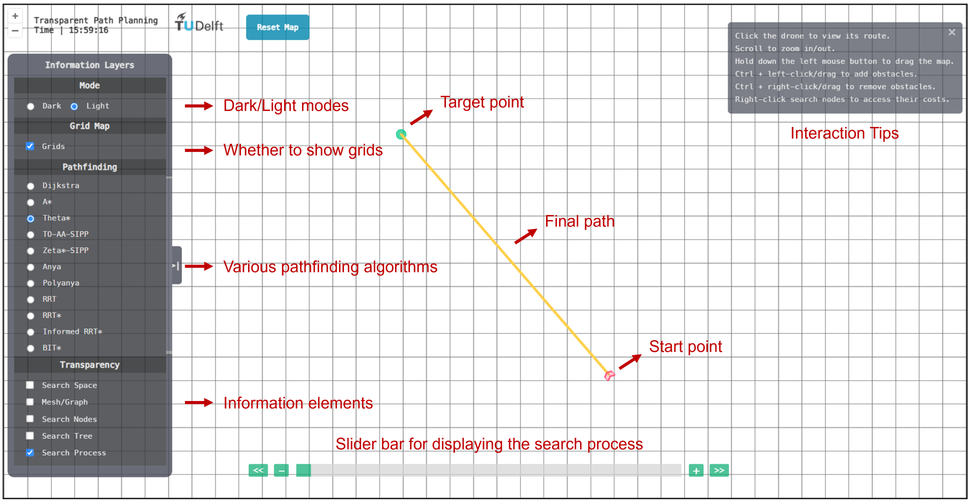


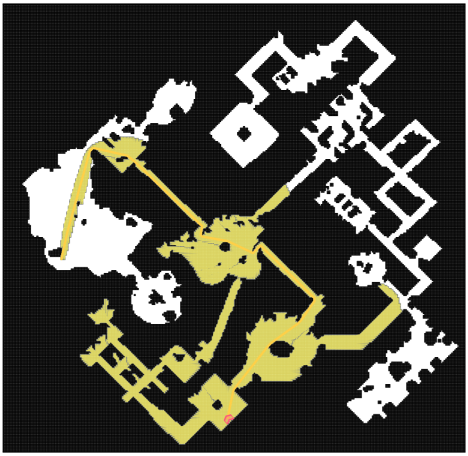
Figure 2.3: The overview of path-planning visualisation.

Essentially, the portrayal of path-planning algorithms can be depicted by search trees comprising both explored nodes and constructed branches, as illustrated in Figure 2.3. While this type of presentation is prevalent for sampling-based path-planning algorithms [46, 47], surprisingly, it is uncommon for grid-based path planning. Generally, the visualisation of grid-based path planning focuses on presenting explored nodes (i.e., grids), neglecting to showcase the expanding, rewiring and pruning processes of search trees

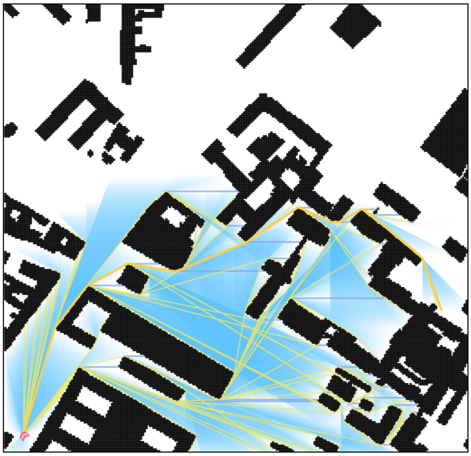
[44, 45]. This research highlights the central importance of search trees in path planning visualisation and algorithmic transparency. By implementing the proposed approach, a new web-based pathfinding visualiser[1] was developed in JavaScript and OpenLayers, as shown in Figure 2.4.



(a) Grid world



(b) AR0602SR (Theta*)



(c) Berlin_0_256 (Anya)

Figure 2.4: Pathfinding visualiser, showing the grid world and two benchmark scenarios.

The visualiser shows the starting position of a generic vehicle (e.g., a drone, car or robot) in a grid world. When the vehicle icon is clicked, its final path and target point will be displayed. The relevant information elements are implemented as five check boxes that follow the top-down structure. Users can activate the boxes to view information on

[1] URL: `http://dronectr.tudelft.nl/`, ID: pathfinder

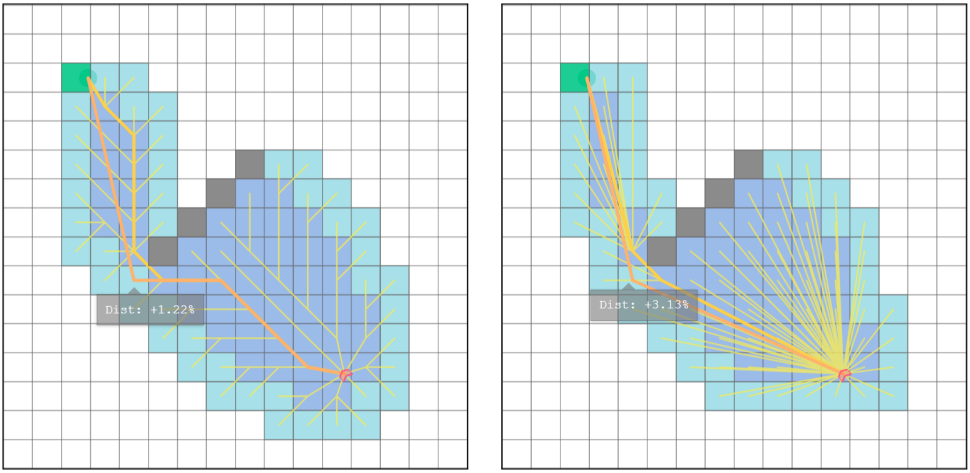
how the selected algorithm works. An interactive slider bar is implemented for displaying search processes, allowing one to progressively step through the entire process either forwards or backwards. Alternatively, the process can be replayed forwards or backwards using the buttons on both sides. Users can freely add or remove obstacles by blocking or unblocking grid elements after which the currently activated path-planning algorithm will generate a new, collision-free path. Note that the grid is independent of the path-planning algorithms and only intended here to help users more easily create and remove obstacles. In addition, the start and target points in the visualiser need not necessarily be situated at the grid centres, which is more applicable to real-world scenarios.

Over ten representative algorithms have been implemented in the visualiser. In this section, six of them are selected to present briefly: A*, Theta*, Anya, Polyanya, Informed RRT* and BIT*. SIPP-based algorithms like TO-AA-SIPP and Zeta*-SIPP are not presented here, as they are designed for dynamic environments. Note that regardless of whether an algorithm is advanced or not, as long as it relies on search trees, it can be visualised using the proposed approach. The nodes of the search trees do not necessarily have to be points; they can also take other shapes, as seen in Anya and Polyanya.

Figure 2.5 shows the search trees and nodes of A* and Theta*. The light blue grids indicate the open nodes, while the dark blue grids denote the closed nodes. The yellow lines are the branches of search trees and the light orange thick line is the optimal path found by the algorithm. It can be clearly seen that A* is limited to 45-degree expansions while Theta* has a straighter and more directly oriented search tree, although their explored nodes are similar. Users can click on any node in the explored space to present an alternative path through the selected node and its corresponding cost. For example, when clicking on the nodes at the same position in Figure 2.5, the distances in A* and Theta* will increase by 1.22% and 3.13% compared to their optimal paths, respectively. The alternative path consists of two parts: the explored path from the start to the selected node and the expected path from the selected node to the target. The explored path follows the branches of the search tree, whereas the expected path reflects the direction to the target. As shown in Figure 2.6, users could also intervene in the output of the algorithm by choosing another (sub-optimal) path present in the stored search tree.

Anya and Polyanya are also A*-based algorithms. Unlike other algorithms, the search node of Anya and Polyanya is a tuple of a root (vertex) and an interval: (r, I), which can be viewed as a triangle as shown in Figure 2.7. The explored space of Anya and Polyanya looks like a collection of rays emitted from a point source. In Anya, the rays radiate up or down row by row, while in Polyanya, the rays diverge polygon by polygon. The optimal path can be found when the target point is covered by the rays. Please note that in Anya, the combination of the triangles (rays) cannot be equivalent to the entire visible region. For example, as shown in Figure 2.7a, the area near the left and right boundaries cannot be fully covered by the rays from the start point. The branches of the search tree in Anya and Polyanya are generated by connecting the roots of the search nodes. The search node creates a continuous region that allows users to click on any point in the explored space to find an alternative path.

Informed RRT* and BIT* are sampling-based algorithms. They both use an *informed* search procedure to narrow the search space and speed up the convergence towards the optimal path. In Figure 2.8, the sampling points are orange whereas the search nodes are



(a) A*

(b) Theta*

Figure 2.5: The search trees and nodes of A* and Theta*.

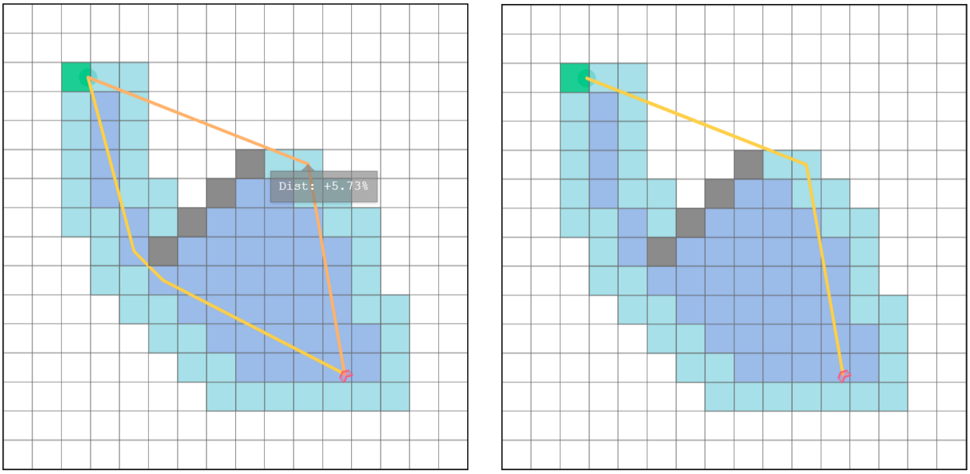
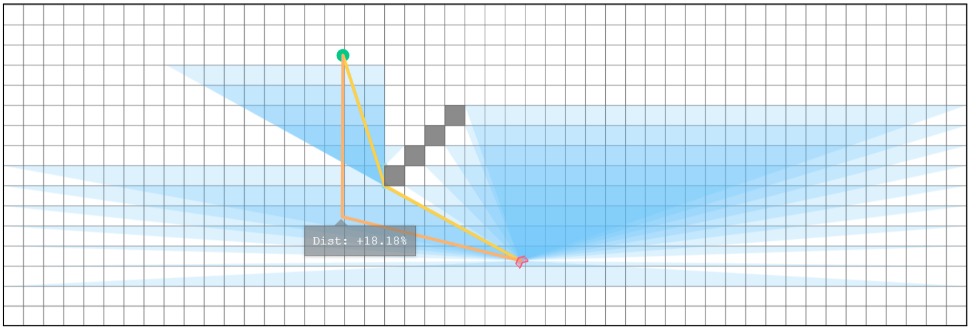


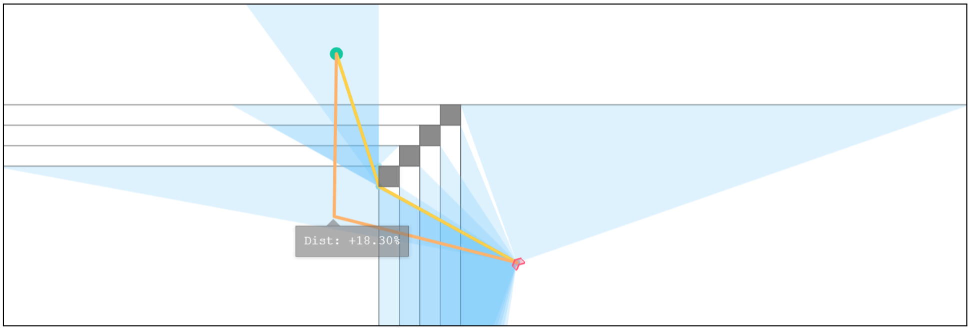
Figure 2.6: Intervening in the Theta* algorithm.

green. They are near-uniformly distributed within the green ellipse, which represents the "current" search space and is defined by the "current" optimal path. An orange circle indicates the rewiring radius of Informed RRT*. Within the rewiring radius, the search tree is re-connected to generate straighter branches. Note that Figures 2.5-2.8 are provided for illustrative purposes only. Readers are encouraged to visit the web-based visualiser[1] to better understand and interpret the information displayed in the figures.

2

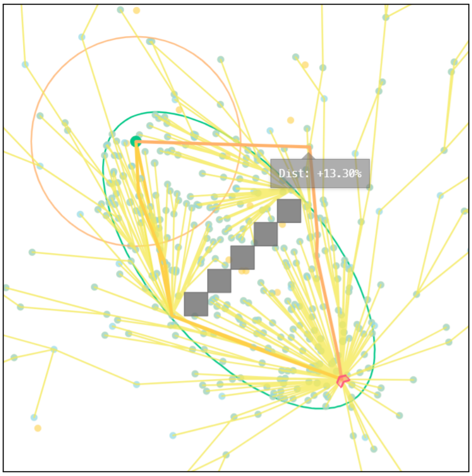


(a) Anya

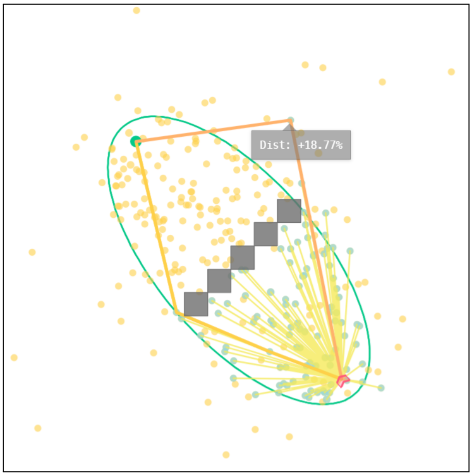


(b) Polyanya

Figure 2.7: The search trees and nodes of Anya and Polyanya.



(a) Informed RRT*



(b) BIT*

Figure 2.8: The search trees and nodes of Informed RRT* and BIT*.

2.4.3. PERFORMANCE ANALYSIS

In this section, the impact of information extraction on the runtime of path-planning algorithms is evaluated. Seven algorithms were chosen: A*, Theta*, Anya, Polyanya, RRT, RRT* and BIT*. Informed RRT* was excluded because, prior to finding the first feasible path, it behaves identically to RRT*, and the sampling-based algorithms were set to terminate upon finding a feasible path. To facilitate comparison, all search trees during the search processes were stored and extracted using both Algorithm 3 and Algorithm 4. The additional runtime introduced by information extraction could thus be analysed.

The experiments were conducted on four public benchmark sets, including around 0.23 million instances [52]. Two of them are from computer games; Dragon Age: Origins and Baldur's Gate II (Original maps). Another one consists of thirty 256 × 256 city/street maps from ten different cities and the final one comprises ten 512 × 512 grid maps, each featuring 10% random obstacles. In the experiments, the maps "den510d" and "orz100d" had to be excluded from the Dragon Age set because the RRT-based algorithms were unable to successfully solve all the problems of the maps within an acceptable time (given fixed parameter settings). For example, the first 1,200 instances of the map "den510d" took RRT 18 days to calculate for both path planning and information extraction on the laptop used for testing. This is because the feasible paths in many instances of "den510d" and "orz100d" need to pass through some extremely narrow passages. The RRT-based algorithms are unable to handle this situation efficiently without optimising their parameters [101, 102]. As the goal is not to compare the performance of different path-planning algorithms, the exclusion of these maps is unlikely to have a significant effect on the conclusion drawn from the benchmark tests.

The algorithms were implemented in JavaScript and performed on Node.js v18.14.2 on a laptop with 2.30GHz Intel Core i7-11800H and 16 GB RAM. For simplicity, rectangular meshes were generated for Polyanya by greedily merging unobstructed grids [82]. The iterations of the sampling-based algorithms were set to infinity, and they were terminated only when finding a feasible path. In this setting, Informed RRT* degenerated into RRT* because its *informed* procedure would be activated only upon the discovery of an initial solution. The BIT* implemented here was its basic version [90].

The performance of sampling-based algorithms heavily depends on their parameter settings. Optimisation of these parameters remains an area of ongoing research [103]. Drawing on the parameter settings from previous studies, the rewiring radius of RRT* and BIT* was computed by $2\eta(1+1/d)^{1/d}(\lambda(X_{free})/\zeta_d)^{1/d}(\log(q)/q)^{1/d}$ [64, 90] where $\eta \geq 1$ is a tuning parameter, d is the dimension of the space, $\lambda(X_{free})$ is the Lebesgue measure of the obstacle-free space, ζ_d is the volume of the unit ball in the d-dimensional space and q is the number of tree nodes. Based on [83, 90], η was set to 1.1. The incremental distance of RRT and RRT* was set equal to the rewiring radius [83] and the target range was 5% of $\sqrt{S}$ where S is the area of the search space [90]. The batch size of BIT* was set to 100 [90].

Tables 2.1-2.3 present the additional runtime introduced by the information extraction methods. To facilitate analysis and comparison, Figure 2.9 summarises the results across the different maps. These results show that, regardless of Algorithm 3 and 4, extracting all search trees from the entire search process could significantly affect the original path-planning runtime, especially for A* and Theta*. This is because A* and Theta*

2

Table 2.1: Mean ratio of additional runtime to original runtime due to `directExtract` of Algorithm 3.

Benchmark	A*	Theta*	Anya	Polyanya	RRT	RRT*	BIT*
Dragon Age	68.9528	26.9459	4.4505	1.8574	0.3315	0.0798	0.0075
Baldur's Gate II	15.0701	6.1994	0.7060	0.3027	0.3476	0.0996	0.0091
City/street maps	23.9722	8.6318	0.6552	0.3663	0.3742	0.0688	0.0069
Random 10%	52.3331	15.5591	10.5738	4.4334	0.4369	0.1326	0.0033

Table 2.2: Mean ratio of additional runtime to original runtime due to recording Γ for Algorithm 4.

Benchmark	A*	Theta*	Anya	Polyanya	RRT	RRT*	BIT*
Dragon Age	0.2002	0.0864	0.0234	0.0089	0.0172	0.0090	0.0016
Baldur's Gate II	0.1725	0.0836	0.0248	0.0071	0.0373	0.0193	0.0030
City/street maps	0.2480	0.1102	0.0202	0.0067	0.0302	0.0114	0.0023
Random 10%	0.1775	0.0929	0.0340	0.0077	0.0022	0.0015	0.0003

Table 2.3: Mean ratio of additional runtime to original runtime due to `forwardExtract` of Algorithm 4.

Benchmark	A*	Theta*	Anya	Polyanya	RRT	RRT*	BIT*
Dragon Age	30.2517	12.6191	2.1679	0.8175	0.1667	0.0640	0.0051
Baldur's Gate II	6.6391	2.8711	0.3197	0.1333	0.1972	0.0790	0.0083
City/street maps	9.4963	4.0004	0.2949	0.1657	0.2168	0.0550	0.0070
Random 10%	21.5124	6.4739	5.6418	2.2755	0.2276	0.0819	0.0015

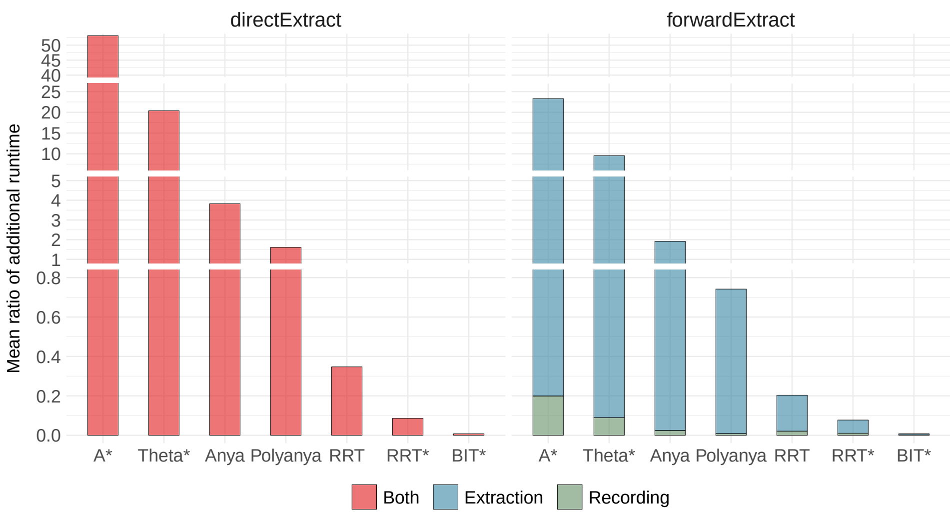


Figure 2.9: Comparison of Algorithms 3 and 4 in data recording and extraction.

both expand search nodes grid by grid and thus construct many branches during their search processes. However, Theta* straightens its search tree by checking if the current node's adjacent nodes can be reached from the current node's parent node and generally constructs fewer branches compared to A* in the same scenario. Therefore, the extraction of search trees has a relatively smaller impact on Theta* compared to A*. In contrast to A* and Theta*, Anya and Polyanya expand their search nodes (triangles) row by row or polygon by polygon, largely reducing the need to build connections (branches) between nodes. Unlike graph-based path-planning algorithms, sampling-based algorithms are less affected by information extraction. This is probably because they are not constrained by predefined graphs and can find feasible paths through fewer branches.

Please note that the results in the tables are also affected by the original runtime. The slower the algorithm, the smaller the impact of extracting the same search tree. For example, RRT* is much slower than RRT due to its rewiring calculations, which results in a smaller impact from information extraction. The mean ratios of additional runtime for BIT* are very small because BIT* consumes considerable time constructing RGGs, which are not considered in the search tree extraction. Further research could combine graph preprocessing with search tree extraction to provide a more comprehensive presentation of path planning. As this chapter focuses on search trees, which are central to path planning, the construction process of graphs is beyond the scope.

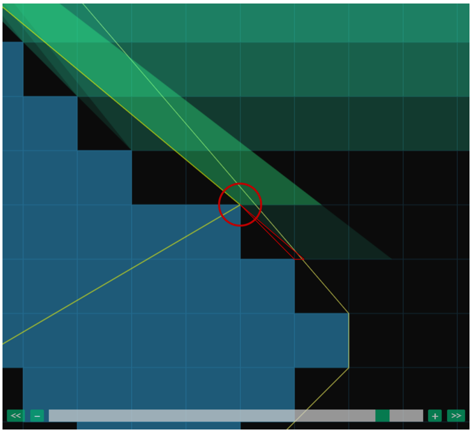
The results of Algorithms 3 and 4 are similar. However, recording only the necessary data Γ will not significantly slow down the original path-planning algorithms. Through Algorithm 4, the search trees can be extracted as needed after the path has been found. The results of this benchmark test also imply that extracting all search trees to present animations may be impractical in real-time, large-scale environments. When computing resources are limited, achieving full transparency could significantly slow down algorithms. As a result, users will receive delayed feedback and information from algorithms, leading to slower responses. In operational contexts with time constraints, this could be problematic and it may be necessary to consider lower levels of transparency (extracting less information). Designing for transparency requires considering both human (cognitive load) and machine (computing resources) limitations.

2.5. DISCUSSION

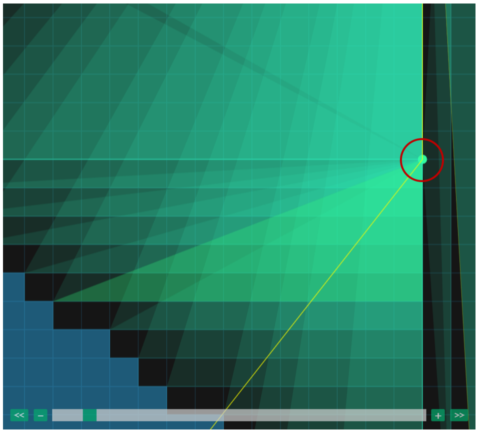
2.5.1. POTENTIAL APPLICATIONS OF ALGORITHMIC TRANSPARENCY

Algorithmic transparency holds numerous potential applications in the real world. For algorithm learners, transparency can facilitate a deeper understanding of path-planning algorithms, making complex concepts more accessible and less intimidating through visualisation [104]. For algorithm designers, it provides a valuable tool for visually debugging incorrect implementations and identifying opportunities for improving algorithms [100]. For system developers, transparency aids in evaluating the strengths and weaknesses of various algorithms, enabling the selection of the most suitable algorithm for a specific application. For operational users, transparency can enhance trust and acceptance of algorithms in operations, fostering productive human-AI collaboration [70].

I have already utilised algorithmic transparency to help me visually debug the implemented path-planning algorithms. For instance, when I tested Anya on the large-scale benchmark map (“Berlin_0_256”), I found some new bugs that did not appear in the simple grid world (Figure 2.4a). In most cases, the final results of Anya were correct and optimal. However, a few scenarios seemed to have some issues: only near-optimal paths were found. I had no idea by only inspecting the output messages on the console. Therefore, I checked their visualisations, with one example shown in Figure 2.10a. In this figure, I found that no new root was generated when the horizontal interval was pushed through a corner (highlighted by a red circle) and thus the true optimal path could not be found. This was because of the precision loss of floating point numbers in JavaScript. It was almost impossible to identify this problem without visualisation. I also tested Polyanya on “Berlin_0_256”, the results were certainly the same as Anya. However, I observed that in several scenarios, Polyanya spent excessive time in finding optimal paths. When one of the scenarios was visualised in Figure 2.10b, it was apparent that redundant nodes were expanded since the turning points should be the corners of obstacles, yet the one highlighted by a red circle was not. Then, by pruning these nodes when generating successors, the runtime of Polyanya could be reduced.



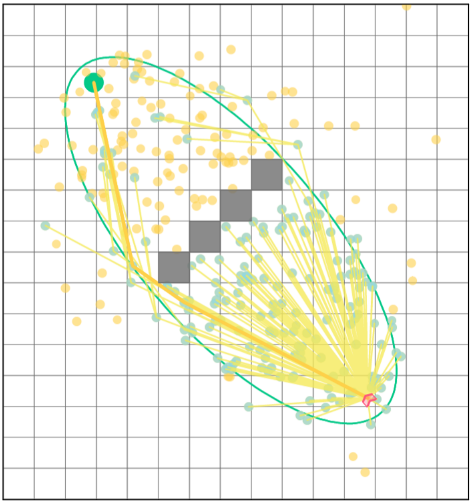
(a) Debugging process in Anya



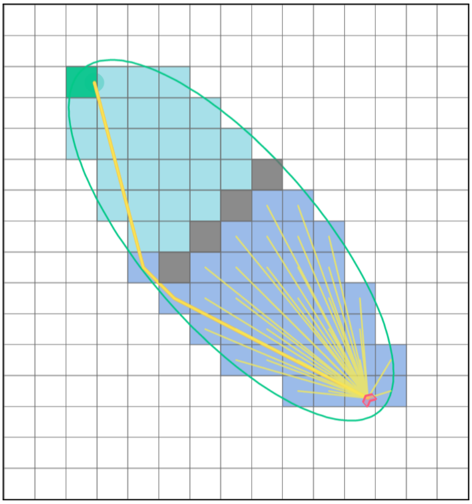
(b) Pruning unnecessary nodes in Polyanya

Figure 2.10: Visualisations for debugging on the benchmark map: Berlin_0_256.

The visualisation also inspired me to improve a path-planning algorithm. During the development process, I found that TO-AA-SIPP inserts too many nodes into the *open* list, with some contributing nothing to finding optimal solutions. This issue seems to arise in sampling-based path planning as well, as it requires generating numerous sampling points and nodes to converge to optimal solutions. Ideally, I only want to sample new nodes that have the potential to improve the current path. This is exactly what Informed RRT* does to RRT*. By observing the visualisations of Informed RRT* and BIT*, I discovered that the “ellipse” they used to limit the search space can be adapted to design an *any-angle forward expansion* for optimal any-angle path planning. This would allow algorithms like TO-AA-SIPP to incrementally add only the necessary nodes to the *open* list, similar to the grid-by-grid forward expansion in A*. The difference is that the grid-by-grid or 2^k-neighbourhood expansion follows a circular pattern, which cannot guarantee finding optimal solutions for any-angle path planning, while the any-angle forward expansion follows an elliptical pattern and can ensure this optimality. By implementing this idea, I successfully improved TO-AA-SIPP and designed Zeta*-SIPP [85], as shown in Figure 2.11. Since there is no dynamic obstacle, Zeta*-SIPP degenerates into Zeta*.



(a) BIT*



(b) Zeta*(-SIPP)

Figure 2.11: Inspiration from the visualisation of BIT* for improving TO-AA(-SIPP).

In safety-critical domains, such as air traffic control, algorithmic transparency is also necessary for supporting human supervision [62]. When automation fails to find a path, operators tend to seek more information to examine the underlying reasons behind the failure [62]. The information overload issue also becomes more evident in tactical operations [19]. In high time-pressure situations, operators may even prefer a “black-box” approach [60]. In this case, a lower level of transparency may be more helpful, as operators may not have time to review the entire search process step by step. It is necessary to develop different levels of transparency that are appropriate for operators to use. Additionally, future research could conduct human-in-the-loop experiments to gather empirical insights into whether algorithmic transparency truly leads to increased understanding of algorithms as expected, particularly among non-experts. It would also be interesting to explore what level of transparency would be required to achieve a basic versus a more thorough understanding of certain algorithms.

2.5.2. THE IMPACT OF INFORMATION EXTRACTION ON ALGORITHM SPEED

From the benchmark tests, it is found that extracting search trees can significantly slow down the algorithms. The fewer the required search steps and branches, the shorter the time needed to extract the information. For example, although the line-of-sight checks in Theta* slow down the runtime, they can result in fewer branches, thereby reducing the need for extraction. Actually, the impact of information extraction depends not only on the number of branches but also on the difficulty of building a branch during the search.

A greater effort in constructing a branch generally suggests a lower ratio of time spent on branch extraction to branch construction. For example, to generate shorter paths on grids, Theta* does not follow predefined edges but performs line-of-sight checks to create more direct paths between vertices. In this case, the effort required to build a branch is greater than that of directly following a predefined edge, thereby indirectly reducing the negative impact of information extraction on algorithm speed. When expanding a branch, RRT-based algorithms have to find the nearest node in the tree to the new sampling point in addition to line-of-sight checks, which further reduces the negative effect.

Although extracting all search trees may be slow, fortunately, the data recording and extraction processes can be separated using Algorithm 4, allowing the latter to be performed in the background after finding paths. As shown in Figure 2.9, the data recording process for Algorithm 4 slows down the original path-planning algorithms by less than 20%, with relatively limited impact on real-time performance of path planning. In practice, the search tree extraction process of Algorithm 4 can be executed in a separate thread to ensure that the main thread, which handles system interaction and path planning, remains unaffected. This can significantly mitigate the negative impact of information extraction on real-time operations.

In general, extracting information from algorithms to achieve transparency will slow down the algorithms anyway. For large-scale operations that demand real-time performance, achieving full transparency may be challenging. Users may experience delayed feedback from the algorithms, leading to slower responses to certain situations and perhaps even instability in closed-loop, human-automation interaction. In that case, lower levels of transparency may be more helpful. For policy makers and algorithm designers, slower algorithm speed may be less of a concern, as they have sufficient time to audit an algorithm. In fact, a slower algorithm might even improve users' assessments [105]. The time spent waiting for the algorithm's output is often used to reflect on the problem at hand and rethink one's own solution, which helps prevent blindly following or dismissing the algorithm's results.

2.5.3. APPLICABILITY AND LIMITATIONS

The proposed visual approach for transparent path planning is founded upon the common steps shared among various path-planning algorithms and can be applied in reverse, as depicted in Figure 2.12. The pathfinding visualiser presented in Section 2.4.2 also illustrates the applicability of the proposed approach.

This research mainly focused on the search process of path-planning algorithms, relatively overlooking the graph construction process. For example, in the empirical analysis, BIT* sometimes consumed much time for building an implicit RGG, but the process cannot be observed through the current visualisation. Although displaying the search process alone is sufficient in many cases as it is the primary process of path planning, it may not be satisfactory for those seeking to uncover every detail.

The proposed approach aims to find a *general* solution for algorithmic transparency in path planning. To achieve that, it is based on the commonalities underpinning the internal process of various algorithms. Thus, it may not incorporate certain computation processes that are specific to certain algorithms. For example, the visibility checks between nodes can be conducted by Line of Sight, but can also be performed by Field of

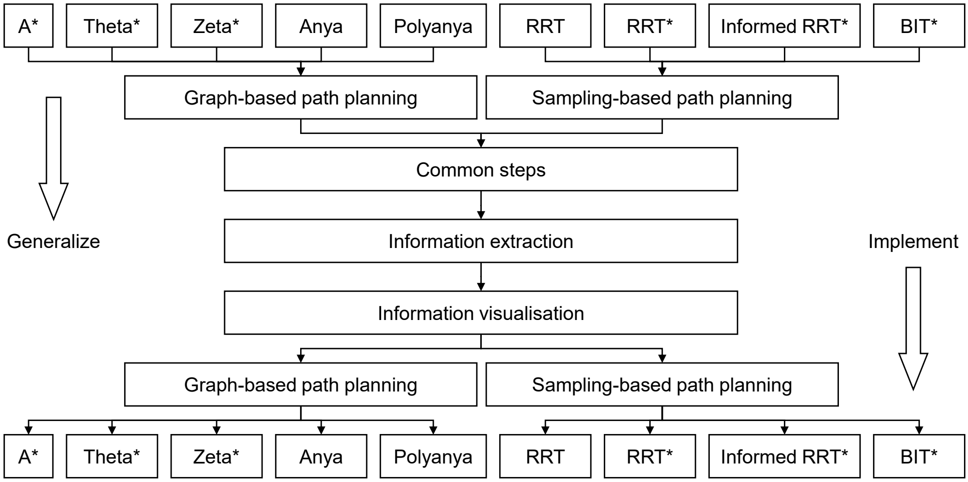


Figure 2.12: Development and implementation of information extraction and visualisation.

View (e.g., Shadowcasting [106]). However, according to the current design, the process of visibility checks remains undisclosed.

In this chapter, no specific application is addressed and thus domain constraints are not yet considered in transparency. In real-world applications, the domain constraints form an essential part of transparent path planning, because it limits the search/solution space [107] and provides additional explanations for why some states or locations cannot be reached or why some solutions are not feasible. To visualise the search/solution space for domain transparency, Ecological Interface Design (EID) [53, 108] could provide some support and reference. It is mainly used to reveal the deeper structure of the work domain that essentially uncovers the operational envelope of physical systems [109]. For example, the search space of flying vehicles is limited by their manoeuvring capabilities and fuel reserves. As such, EID could be used to complement this research.

2.6. CONCLUSION

This chapter presents a visual approach to transparent path planning, revealing the internal processes of path-planning algorithms by putting the emphasis on search trees. For graph-based and sampling-based path planning, two information extraction methods are introduced based on the common steps shared by various algorithms. To demonstrate the proposed approach, a novel web-based pathfinding visualiser has been developed in JavaScript and OpenLayers. Over ten representative path-planning algorithms have been implemented in the visualiser and six of them are discussed in detail. It is anticipated that the visualisations enable users to understand the algorithms and perceive their strengths and weaknesses.

Benchmark tests indicate that information extraction can have a large negative impact on algorithm runtime, particularly for A* and Theta*. Two strategies can be applied to mitigate this negative impact: 1) Information extraction can be separated from data recording and performed after the path is found using Algorithm 4. 2) The transparency level can be lowered to minimise the amount of information that needs to be extracted. The second strategy requires a transparency model, which will be developed in the next chapters. In this chapter, domain constraints were ignored since no specific application is specified. Future research can integrate algorithmic transparency with domain transparency towards transparent path planning in operational contexts. This will be explored in Chapters 4 and 5 for Uncrewed Air Traffic Management (UTM).

3

Impact of transparency on understanding

In the previous chapter, a visual approach to transparent path planning was introduced, revealing the search processes of path-planning algorithms. This chapter further organises the path-planning transparency into distinct levels and then explores their impact on human understanding via a user study. The results show the effectiveness of transparent path planning and demonstrate that increased transparency could lead to enhanced understanding in path planning.

The contents of this chapter are based on:

Paper title	Algorithmic transparency in path planning: A visual approach to enhancing human understanding
Authors	Yiyuan Zou and Clark Borst
Published in	International Journal of Human-Computer Studies **203**
DOI	10.1016/j.ijhcs.2025.103573

ABSTRACT

Computer algorithms facilitate increased automation in various human-centred work areas to improve operational safety and efficiency. Algorithmic transparency is considered essential for human operators, policy makers and system developers, as it allows them to understand the capabilities and limitations of an algorithm. However, it remains unclear to what extent algorithmic transparency can enhance human understanding, particularly for path-planning algorithms. Therefore, this chapter describes results of a user study among non-experts ($N = 40$) to evaluate the impact of algorithmic transparency on human understanding in path planning. Drawing on theories from cognitive engineering and the notion of progressive disclosure, six levels of transparency were designed to chunk meaningful information pertaining path-planning algorithms. During the study, transparency information was progressively disclosed in levels, and participants' understanding at each level was assessed through objective questions and subjective confidence ratings. The results suggest that increased transparency levels allow non-experts to more correctly and confidently understand the details of a path-planning algorithm. However, it is also found that certain transparency levels can lead to confusion, especially when the algorithm behaves in a way contrary to human expectations. This study further reveals that, given the same level of transparency, sampling-based path-planning algorithms may be easier to comprehend than graph-based algorithms. This research can serve as a reference for how to hierarchically portray and organise transparency information and how to implement transparency in path-planning applications.

3.1. INTRODUCTION

As technology advances, more and more algorithms are being developed to help people do their jobs more safely and efficiently. However, the complexity of algorithms is also increasing, making them more difficult for humans to understand. This lack of understanding hampers human trust and acceptance of the advanced technology, thereby limiting its real-world applications, especially in safety-critical domains [33, 110, 111]. In aviation, operator acceptance has proven to be one of the largest challenges to successfully introducing new advanced automation [112, 113].

To address this issue, Explainable Artificial Intelligence (XAI) has emerged in recent years, attracting widespread attention from researchers in various fields [31, 32, 36, 70, 114, 115]. XAI is defined as "*AI systems that can explain their rationale to a human user, characterise their strengths and weaknesses, and convey an understanding of how they will behave in the future*" [31, p. 44]. It was originally derived from Explainable Machine Learning (XML), as advanced ML models, especially Neural Networks (NN), are generally too complicated to interpret [116, 117]. Researchers seek to comprehend the knowledge embedded within a trained ML model or the decision rules the ML model acquires via learning.

Nowadays, XAI has become a multidisciplinary field [111, 114, 115], covering AI [93, 118], Human-Computer Interaction (HCI) [21, 119] and Cognitive Science [120, 121]. This expansion is driven by the recognition that understanding AI is beneficial to a wide range of stakeholders, including not only researchers, but also engineers, designers, domain users, and regulators [33, 111]. The European Union (EU) General Data Protection Regulation (GDPR) has established the "Right to Explanation" for individuals affected by algorithmic decisions [35]. Since AI is a broad topic, this research focuses mainly on path planning, exploring a practical approach to enhance its comprehensibility.

Some initial efforts have been made in this direction, such as Explainable AI Planning (XAIP) [73, 74, 122] and Explainable Motion/Path Planning [71, 72, 98, 123, 124]. *Model Reconciliation* [74, 122] is one of the representative works in this field. It was developed by considering the process of explanations as the reconciliation between human mental models and AI models. There, the model difference, along with inferential limitations of humans, is the root cause of the need for explanations [73]. However, this approach heavily relies on estimating human mental models [125, 126], which may generate meaningless explanations if the mental models are inaccurately estimated.

To complement the model reconciliation, "seeing-into" *transparency* [19–22, 30] are therefore considered. Rather than identifying the model difference, transparency entails revealing the internal processes of AI models, making them accessible to humans. By reviewing the transparency information and learning from it, humans can also narrow the gaps between their mental models and the AI models. The advantage of transparency is that it gives users access to all relevant information. The downside, however, is that too much information can become overwhelming and impede understanding.

To avoid overwhelming human users, especially in operational contexts, many studies recommend organising transparency information hierarchically and providing information on demand [19, 22, 30, 62]. Therefore, rather than just animating algorithms, as existing pathfinding visualisers do [44, 48], six levels of visual transparency were devised for path planning based on Chapter 2. A user study was then conducted to evaluate the impact of different transparency levels on human understanding.

Relevant research regarding the effects of algorithm visualisation/animation on understanding has been performed in Computer Science (CS) education [40, 104, 127, 128]. However, in CS education, algorithm visualisation mainly serves as a supplementary or auxiliary material to the core textbook teaching materials. In this research, the primary interest is in the level of understanding achievable solely through visualisation. If a low level of transparency can lead to a relatively high level of understanding, perhaps a small amount of information is sufficient to meet user needs, thereby reducing the possibility of information overload in operational scenarios. In the user study, non-experts were invited to participate, since experts with prior knowledge of path planning may confound the results. Moreover, users of path-planning applications may be the ones who are unfamiliar with path-planning algorithms, such as air traffic controllers and policy makers. The study results could potentially provide a reference for these use cases.

Combined with Chapter 2, this research could serve as a practical example of operationalising a multidisciplinary perspective to study XAI and transparency [111, 114, 115]. While this research specifically addresses path planning, its methodology can be generalised to other AI fields for promoting the real-world applications of AI [129].

This chapter is structured as follows: Section 3.2 reviews relevant literature on transparency design frameworks and guidelines, and user studies. Section 3.3 shows the design of six levels of visual transparency for path planning. Sections 3.4 and 3.5 present a user study and its results for evaluating the impact of these transparency levels on human understanding. Section 3.6 discusses the findings and limitations of this research.

3.2. RELATED WORK

3.2.1. DESIGN FRAMEWORKS

From a user-centred perspective, explanatory and transparency information should be presented in accordance with user demands, limitations, preferences and needs. Several user-centred frameworks have been proposed to design XAI systems [97, 114, 130, 131]. Wang et al. [131] proposed a theory-driven user-centric XAI framework, inspired by the theoretical underpinnings of human reasoning. This framework aims to support human reasoning processes and reduce cognitive biases by providing tailored explanations. Instead of abstract design guidelines, Eiband et al. [130] introduced a practical approach with their stage-based participatory design process for achieving transparency by explanation interface design. This process can be divided into two pivotal phases: identifying the content of explanations and determining the best approach to deliver them, thus offering concrete guidance for designers. Building upon these foundations, Mohseni et al. [114] developed a nested XAI framework that integrates both design and evaluation phases. This framework contains three layers: an overall XAI system, an explanation interface, and core explainable models and algorithms. The design process of this framework initiates with defining the goals of the XAI system in the outer layer, progresses by tailoring the interface to meet user needs in the middle layer, and finally focuses on the underlying algorithms in the innermost layer. In each layer, the stages of design and evaluation construct an iterative cycle.

Another line of research is related to automation and agent transparency [19], such as the Belief-Desire-Intention (BDI) framework [132], Lee and See's 3Ps (Purpose, Process and Performance) theory in human trust in automation [133], Lyons' human-robot transparency model [21] and the Situation Awareness-based Agent Transparency (SAT) model [22]. To avoid overwhelming human users, transparency is generally divided into different levels, enabling a progressive and incremental disclosure of information [30]. For example, the SAT model has three levels [22, 134]: Level 1 - Basic Information (agent's current status/actions/plans), Level 2 - Rationale (agent's reasoning process) and Level 3 - Outcomes (agent's projections/predictions; uncertainty), each corresponding to and supporting the three levels of Endsley's Situation Awareness theory respectively [135]. More relevant reviews are provided by Bhaskara *et al.* [19], Endsley [70], van de Merwe *et al.* [136].

Although many transparency frameworks were developed for multi-agent systems, they can also serve as a reference for the algorithmic transparency design. These frameworks often divide transparency into different components and organise them into a hierarchical structure. As deeper information is revealed, a higher level of understanding could be achieved. The research presented in this chapter also explores how to divide path-planning transparency into different elements and how to design transparency levels to progressively disclose information about path-planning algorithms.

3.2.2. USER STUDIES

To evaluate the impact of transparency on user perceptions, understanding, acceptance and trust, some user studies and human-in-the-loop experiments have been conducted [19, 114, 136, 137]. However, the reported study results are mixed [30, 138]. Some studies show clear positive effects of transparency [24, 34, 139–142] while others do not [143, 144]. These findings indicate that additional factors influence the effectiveness of transparency, such as explanation modalities [145], explanation meaningfulness [146], information amount [144, 147], user expectations [30, 148, 149] and domain knowledge [150]. Designing for transparency should take these factors into account. Interestingly, investigations in CS education have also identified similar mixed results regarding the effectiveness of algorithm visualisation [151]. Through a meta-study [40] and a series of experiments [152], researchers found that *engagement* (e.g., interacting with the algorithm) plays a vital role in learning and understanding [127, 152].

To assess the proposed approach for transparent path planning, a user study is also required. Considering the central importance of understanding in transparency [111], the user study in this chapter was specifically designed to explore the impact of transparency on understanding rather than trust and task performance. Moreover, previous user studies have rarely assessed the understanding of planning processes, focusing primarily on classification and prediction problems [145], or planning outputs (final plans) [74]. This study could fill this gap and serve as an example of how to conduct such an evaluation. To avoid potential bias from experts' prior knowledge, only non-experts were invited to participate. An interactive interface was implemented to enhance participant engagement [142], allowing them to freely explore the algorithm's behaviour and maximise the use of available information at different transparency levels.

3.3. TRANSPARENCY LEVELS

For the purposes of learning and auditing, it may be helpful to have browsable animations that offer comprehensive insights into the inner workings of path-planning algorithms. However, in operational contexts, presenting excessive information simultaneously may overwhelm operators [19]. As suggested by Springer and Whittaker [30], progressive disclosure may be necessary for algorithmic transparency, indicating that transparency information needs to be organised hierarchically from 'simple' to 'complex'.

No clear guidelines exist, however, for what type of hierarchy would be needed and what 'simple' and 'complex' entail. Therefore, to clarify algorithmic transparency in path planning, the algorithms are decomposed into multiple information elements, which are then categorised into six levels, as shown in Figure 3.1. Inspired by Rasmussen's Abstraction Hierarchy (AH) used in Cognitive Systems Engineering and Ecological Interface Design (EID) [53, 153, 154], the transparency levels were organised from functional purpose (i.e., solution) to inner physical structure (i.e., search process), following a typical human top-down, problem-solving strategy. As the level of transparency increases, more details about the algorithm's inner structure are revealed in a cumulative way. Each subsequent, deeper level introduces additional information on top of the preceding levels without excluding any previously disclosed information.

A general example of the transparency levels for two different types of path-planning algorithms is shown in Figure 3.2. At Level 0, only the final path is presented. The dashed line indicates the direct path without obstacles. The final path will cause a 5% additional cost compared to the dashed direct path (the numbers are illustrative only). At Level 1, the search space is delineated by an elliptical region, which could be more complex in real-world scenarios (more constraints). Here, the elliptical region can represent the

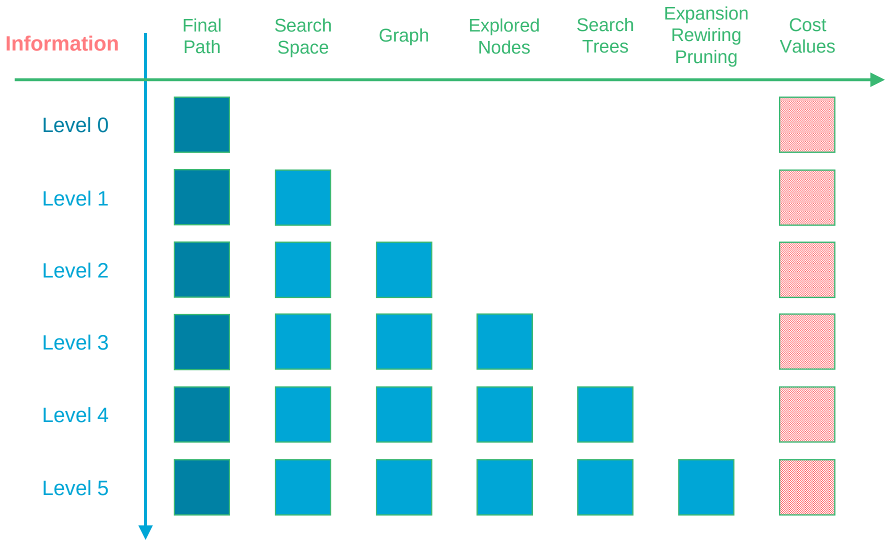


Figure 3.1: Proposed transparency levels for organising different information elements in path planning.

endurance/fuel or required time of arrival for moving vehicles [107]. At Level 2, regular grids and sampling points that discretise the search space are showcased. In Probabilistic Roadmap (PRM)-based algorithms, the connections between sampling points can also be highlighted. At Level 3, the explored space is illustrated through the display of explored nodes. Users are able to probe the explored space, for example by clicking on a node, to obtain a cost-based contrastive explanation [115] for why the optimal path is A rather than B. Please note that this click will not trigger a new search but will retrieve the explored path through the clicked node from the search tree. At Level 4, the search tree is shown to provide detailed information about the explored solution space. From the figure, one can see many other feasible solutions besides the optimal one. It also provides users with the possibility to change the optimal path to a non-optimal, but maybe a more preferable path, allowing users to intervene in the algorithm based on the existing information [114, 123].

Level 5 is omitted from Figure 3.2 because the transparency elements regarding the search process are designed to be displayed via a controllable animation. Animation, as a dynamic visualisation technique, has been shown to be helpful in explaining algorithmic behaviour and improving human understanding [41, 155]. According to the cognitive load theory [156], the capacity of humans to process novel information is limited. Animation could make hidden dependencies among different components (e.g., nodes and branches) in a model more salient and thus reduce the cognitive load associated with comprehension [41]. Animation is quite suitable for presenting time-related procedural knowledge [40–42]. Through animation, the dynamic expansion, rewiring, and pruning of search trees can be visually observed. Moreover, animation should be both controllable and interactive given the complexity of the information it conveys [42]. Users should

3

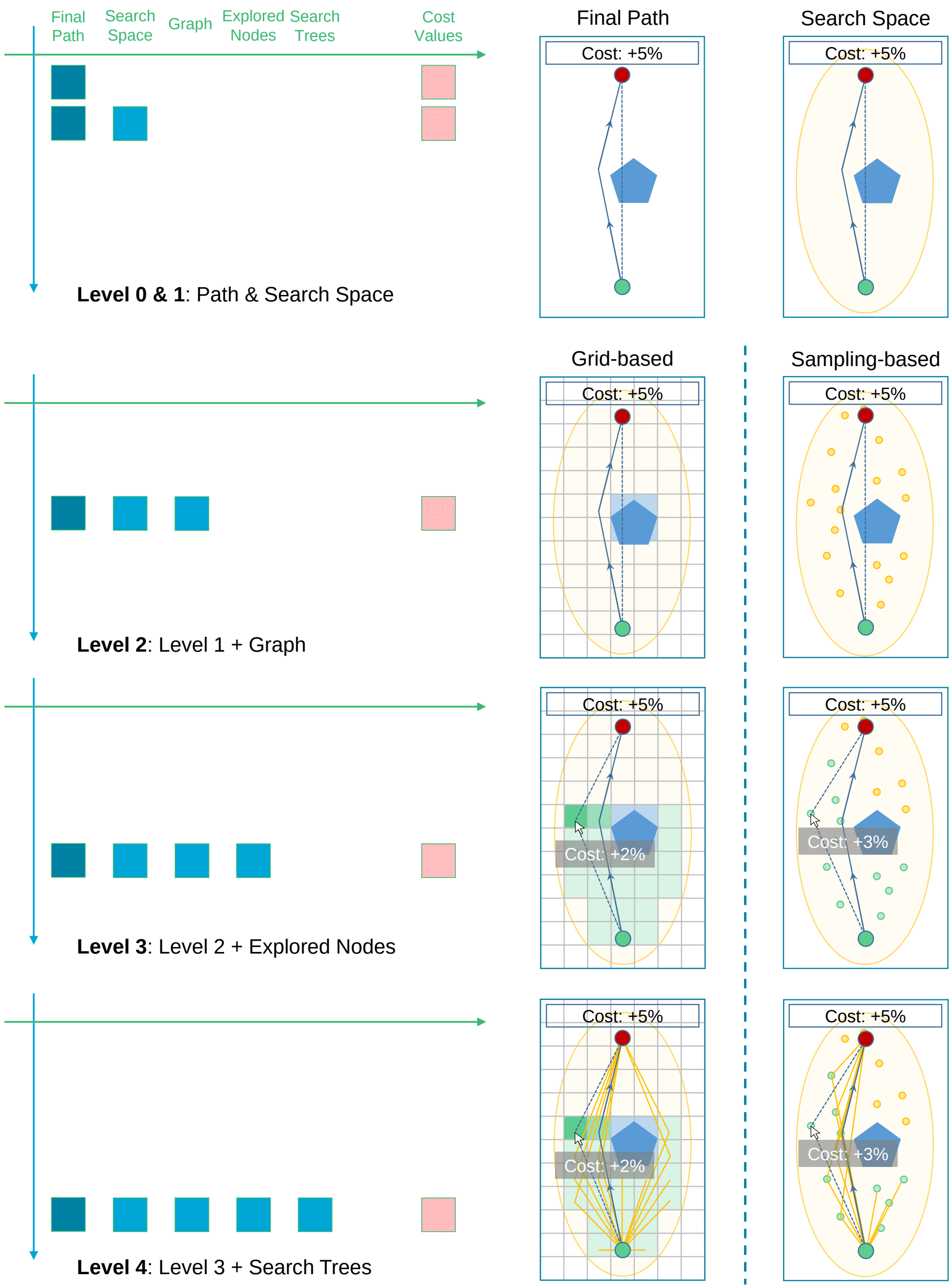


Figure 3.2: Implementation of the transparency levels.

have the flexibility to view and review, stop and start the animation.

Figure 3.2 actually hints at another reason for adopting the cumulative approach in classifying transparency levels: the transparency elements are not completely independent of one another. For example, the graph can be regarded as a discretised version of the search space, allowing one to roughly perceive the shape, size and boundary of the original search space. The search trees contain the location information of the explored nodes and are also shaped by the graph and search space. The search process can be regarded as a sequence of search trees, with the final tree representing the final result of the explored solutions. Therefore, the new information introduced at deeper levels is actually dependent on the previous levels, but further elucidates the algorithm's inner workings in detail. For an expert familiar with the algorithm, some information at lower levels can be hidden to avoid visual clutter without compromising comprehension.

3.4. METHODOLOGY

This section presents a user study evaluating the impact of algorithmic transparency on human understanding. Its goal was to gain empirical insight into how transparency enables *non-experts* in path planning to correctly and confidently understand details of a path-planning algorithm. The underlying motivation for this choice is the expectation that end users, system developers/engineers, and policy makers without a computer science background will probably need to work with such algorithms in their professions.

3.4.1. EXPERIMENT SETUP AND PROCEDURE

The experiment was conducted in a laboratory at TU Delft using the developed pathfinding visualiser, as shown in Figure 3.3. As the laboratory is an enclosed room, only one participant was allowed at a time. The task for participants was to understand the underlying path-planning algorithm through various transparency levels. As shown in Figure 3.4, in the experiment, the scenario remained consistent across all participants, algorithms and transparency levels, with fixed start and target points and modifiable obstacles. Participants were allowed to freely add or remove static obstacles by blocking or unblocking grid elements.

The experiment consisted of two main phases: training and measurement. The domain constraint considered in the measurement phase was that the path length cannot exceed a certain value. This constraint is very common when a vehicle needs to reach its destination within a limited time (e.g., battery power or emergency). Two different types of path-planning algorithms were chosen: Theta* and Informed RRT*. Each participant was assigned one of them for evaluation, following a between-participants design. During the experiment, participants were presented with different levels of transparency, starting from Level 1 to Level 5. At each level, participants were required to answer questions designed to test their understanding.

Considering that maintaining focus for a long time is difficult, the entire experiment generally took 90 minutes, with the measurement phase having a maximum time limit of 60 minutes. Figure 3.5 shows an example of the interface (Theta*, Level 5) presented to participants[1]. Level 0 was excluded in the experiment according to three rounds of

[1] URL: `http://dronectr.tudelft.nl/`, ID: understanding

Figure 3.3: Experiment setup in the laboratory.

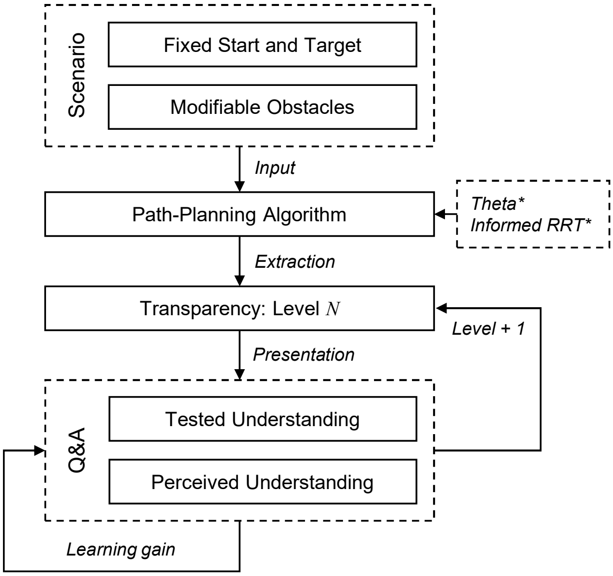


Figure 3.4: Overview of experimental design.

beta tests. During the beta tests, ambitious and exploration-minded participants found it difficult to complete the experiment containing Level 0 on time. Given that Level 1 only reflects the domain constraints imposed on the algorithms, the information it contains is relatively independent of algorithmic information. Hence, setting Level 1 as the baseline

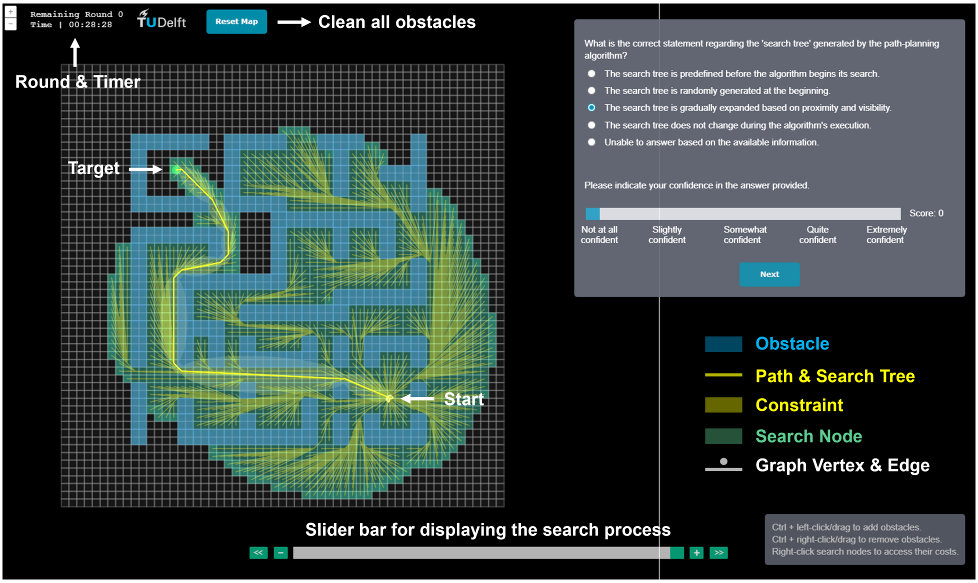


Figure 3.5: Screenshot of the experiment interface (Theta*, Level 5).

is also appropriate.

The questions to be answered included objective and subjective types, corresponding to tested and perceived understanding respectively. The objective questions were mostly multiple-choice, each with three incorrect choices and one correct choice. Each question had an “Unable to answer based on the available information” option, which participants were encouraged to select only if they felt they lacked sufficient information to answer accurately. The questions were devised based on the transparency elements in Figure 3.1. According to DARPA’s XAI definition [31], users should be able to understand the weaknesses of AI systems via XAI. Thus, participants were also asked about the disadvantage of the presented path-planning algorithm. When submitting an answer to an objective question, participants were asked to indicate their confidence in their responses (from 0 to 100). By considering both tested and perceived understanding, one’s calibrated understanding could be assessed [157]. Well-calibrated understanding indicates high confidence in correct answers and low confidence in incorrect ones. The final question was open-ended, designed to elicit participants’ explanations regarding the inner workings of the algorithm.

After all questions were answered and submitted, the transparency level would automatically increase to the next level. At the next level, participants could either adjust their initial answers or submit the same answers. This process continued until the end of the highest level of transparency. To mitigate learning effects in the experiment, the ordering of the objective questions was randomised for each participant.

The eight objective questions were shown as follows, with more details available in Appendix B. These questions could also be answered through textual and/or verbal explanations, similar to those provided by textbooks and teachers. However, the focus was

specifically on how visual transparency would enable participants to answer them.
Q1: The algorithm will always find the same path in the same situation [True or False].
Q2: The algorithm will always find the true shortest path (if a path exists) [True or False].
Q3: What is the most accurate statement regarding the constraint for the final path?
Q4: What is the main advantage of the discretisation method adopted by the algorithm?
Q5: What is the correct statement regarding the search nodes explored by the algorithm?
Q6: What is the correct statement regarding the search tree generated by the algorithm?
Q7: What strategy does the algorithm employ to attempt to find the shortest path?
Q8: What is the main disadvantage of the algorithm?

Before the experiment began, participants were required to undergo training to familiarise themselves with the interface elements, interactions, and questions along with their options. Dijkstra's algorithm [49] was implemented and presented in this phase. The constraint and grid size were both different from those used in the measurement phase. For illustrative purposes, the transparency elements in Figure 3.1 were presented in random combinations during training, rather than at the proposed levels.

Additionally, participants were not informed of the presentation order of the transparency levels in the measurement phase. They only knew the number of the remaining rounds they needed to complete. This setup made it difficult for participants to predict what the supportive information would be available next, preventing them from intentionally skipping to higher levels to gather more information for answering questions.

Participants were encouraged to answer the questions to the best of their ability using the information available to them. After the experiment, participants were asked to complete a post-hoc questionnaire in which they could indicate their preferences regarding the transparency elements using a five-point Likert scale.

3.4.2. PARTICIPANTS

Forty participants, all TU Delft staff and students from faculties Aerospace Engineering, Civil Engineering and Geosciences, Mechanical Engineering, and Technology, Policy and Management, volunteered to conduct the experiment. All of them had a general understanding of path-planning problems, but lacked in-depth knowledge of pathfinding algorithms. In fact, expert knowledge of pathfinding algorithms could bias or confound results, making it difficult to attribute understanding to the portrayed information. Participants were quasi-randomly assigned to the groups to create two balanced groups based on education level (see Figure 3.6). This experiment was approved by the Human Research Ethics Committee (HREC) under number 4019.

3.4.3. INDEPENDENT VARIABLES

The experiment had two main independent variables: 1) two path-planning algorithms (between-participants), and 2) five transparency levels (within-participants). Theta* and Informed RRT* were chosen because they are the classic (and still advanced) examples of graph and sampling-based algorithms. They also have different implications for algorithm visualisation, and thus potentially understanding. For each participant, the transparency levels all started at Level 1 and automatically increased to the next level after all questions were answered.

Reverse or random ordering of the transparency levels was not adopted as a compar-

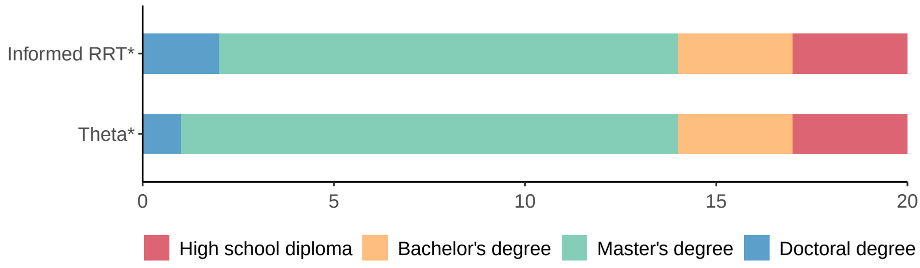


Figure 3.6: Distribution of the participants' education levels.

ison group because the higher levels always contain information from the lower ones, potentially influencing the learning outcomes of the lower levels. Moreover, the integration of information at the higher levels can circumvent the phenomenon of forgetting, ensuring that any enhanced understanding is attributable to the new information.

3.4.4. DEPENDENT MEASURES

Four types of dependent measures were considered: 1) understanding, 2) learning time, 3) interactions, and 4) preferences. Participant understanding was measured in terms of tested and perceived understanding [114, 157, 158]. The tested understanding represented the number of correct answers to the objective questions whereas the perceived understanding was measured by the participants' self-rated confidence. The learning time was recorded by the time it took participants to answer each question at each transparency level. To avoid learning effects in the experiment, the objective questions were presented in a random order. The interaction metrics include the number of algorithm executions and interactions with the map (e.g., grid activation and deactivation). Participants' preferences regarding the usefulness of the transparency elements (for the purpose of understanding) were measured on a five-point Likert scale.

3.4.5. CONTROL VARIABLES

There were five control variables: 1) interaction mode, 2) questions and options, 3) the order of transparency levels, 4) algorithm parameters, and 5) start and target points.

At each transparency level, participants could freely interact with the path-planning algorithm by adding or removing obstacles, allowing them to observe how the solutions could be affected. This method can be regarded as *discovery learning* [159, 160] or *active learning* [161], which emphasises the role of engagement in learning and understanding [40, 127, 152]. In this manner, participants could maximise the use of available information to comprehend the algorithm, allowing for a more accurate assessment of the impact of algorithmic transparency on human understanding. To maintain consistency, the scenario would be reset to its initial condition, removing all obstacles, before participants began answering questions at each level.

Additionally, as indicated by [162], guided discovery is generally more effective than pure discovery in learning. Therefore, the questions and their options were integrated into the interface, rather than being added as a post-hoc quiz, to serve as guidance or learning objectives for participants, similar to an “open-book” exam. These questions and options remained consistent across all algorithms and levels of transparency. Only the correct answers to the questions may vary for different algorithms.

As mentioned above, the order of transparency levels was fixed, ranging from Level 1 to Level 5. The algorithm parameters might also influence participants’ comprehension as the algorithm’ behaviour may differ depending on the parameters, such as the grid size in Theta* and the incremental distance in Informed RRT*. To facilitate comparison, the algorithm parameters, along with the start and target points, were kept fixed as well. For Theta*, the grid size is set to half the size of the obstacle grid. For Informed RRT*, the maximum number of iterations is set to 1000. To make the algorithm’s behaviour more apparent, the rewiring radius is set to $d_{st}/3$ and the incremental distance to $d_{st}/5$ where d_{st} is the distance between the start and target points.

3.4.6. HYPOTHESES

It was hypothesised that a higher level of transparency would lead to a deeper understanding of path-planning algorithms, indicating higher fractions of correct answers and higher self-rated confidence scores (H1). It was also expected that Informed RRT* would be comparatively more challenging to comprehend than Theta* (H2). This means that Theta* may be grasped more quickly (less learning time) and effectively (higher learning scores) than Informed RRT* at the same level of transparency. This hypothesis was driven by the observation that the behaviours of sampling-based algorithms are more difficult to predict because of their random sampling strategies. In contrast, graph-based algorithms are generally more structured and deterministic.

3.5. RESULTS

Table 3.1 shows the means and standard deviations of the dependent measures at different transparency levels for Theta* and Informed RRT*. In general, as the level of transparency increases, both the hit ratio and the confidence increase, while the learning time and the number of interactions (algorithm executions and interactions with the map) tend to decrease. Surprisingly, the hit ratio of Theta* is lower than that of Informed RRT*, contrary to the hypothesis (H2). The learning time of Theta* is generally lower than that of Informed RRT* at the same level (except for Level 1), aligning with the expectations. The following sections will further discuss the results in more detail.

3.5.1. DATA ANALYSIS AND STATISTICS

To compare Theta* and Informed RRT*, the means of dependent measures for each participant were first computed, and then Mann-Whitney U tests were conducted for statistical analysis. The effect size r was calculated in the Mann-Whitney U tests (small ≥ 0.1, medium ≥ 0.3, large ≥ 0.5), which is defined by the standardised test statistic z from the tests divided by the square root of the total number of observations. To compare different transparency levels in each algorithm group, the dependent measures were analysed using Friedman tests, followed by Exact tests [163] with Bonferroni correction for further pairwise comparisons. Kendall’s coefficient of concordance w was used to measure the effect size for the Friedman tests (small ≥ 0.1, medium ≥ 0.3, large ≥ 0.5). The significance

Table 3.1: Means of Dependent Measures (DM) as a function of transparency level.

Algorithm	DM	Level 1	Level 2	Level 3	Level 4	Level 5
Theta*	Hit Ratio	0.32 (0.17)	0.42 (0.21)	0.44 (0.16)	0.49 (0.13)	0.54 (0.17)
	Confidence	58.5 (13.8)	63.2 (17.6)	66.2 (17.1)	70.3 (17.8)	75.2 (18.3)
	Learning Time	774 (276)	526 (141)	552 (217)	491 (242)	474 (231)
	Execution	63 (39)	39 (27)	40 (25)	28 (16)	22 (22)
	Map Interaction	257 (169)	140 (88)	123 (97)	68 (56)	77 (71)
Informed RRT*	Hit Ratio	0.47 (0.18)	0.53 (0.19)	0.57 (0.16)	0.61 (0.12)	0.63 (0.14)
	Confidence	57.3 (13.8)	62.4 (13.3)	65.3 (13.8)	69.5 (14.7)	72.8 (16.6)
	Learning Time	769 (237)	612 (220)	560 (193)	531 (182)	674 (288)
	Execution	109 (119)	69 (63)	74 (76)	63 (63)	34 (43)
	Map Interaction	244 (132)	145 (104)	137 (114)	110 (101)	92 (85)

Note: Standard deviations are in parentheses.

level was set to 0.05. As the effect size reflects the magnitude of the difference between groups [164], it is ideal to have both a statistically significant result ($p < 0.05$) and a large effect size to claim a clear and meaningful difference. A low effect size suggests that the statistical significance should be interpreted with caution.

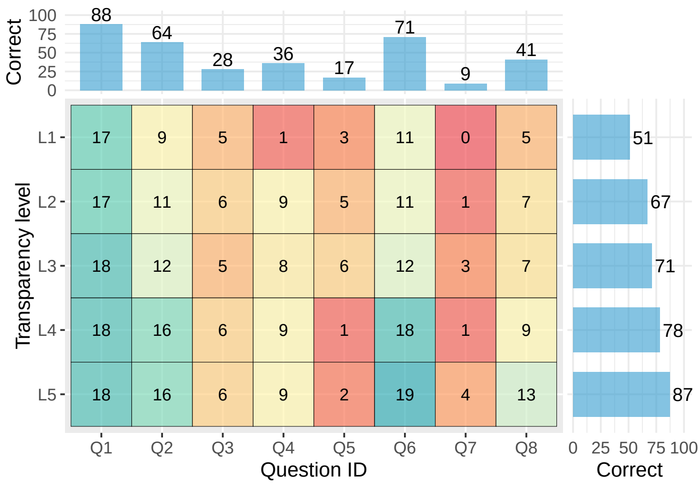
3.5.2. HIT RATIO

Figure 3.7 shows the number of correct answers for each question at each transparency level. The bar plot above the heat map illustrates the cumulative number of correct answers for each question, whereas the bar plot on the right indicates the cumulative number of correct answers for each transparency level. It is evident that some questions are difficult to answer correctly, while others are not.

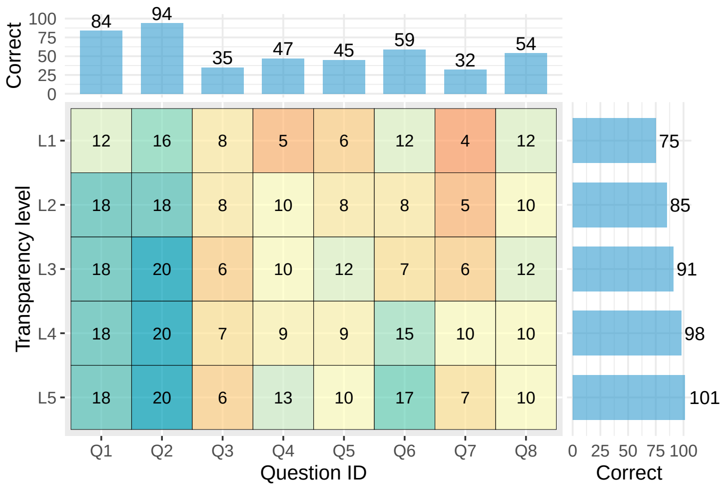
The first question, Q1, asks whether the algorithm is deterministic or probabilistic. Most participants answered this question correctly across all transparency levels. However, several participants in Informed RRT* relied too much on their expectations of the algorithm and failed to notice that it generated different results in the same environment, leading to incorrect answers to Q1 at Level 1. Fortunately, when they observed the sampling points at Level 2, they correctly revised their answers.

Q2 inquires whether the algorithm can find the true shortest path. As the path found by Theta* is close to the true shortest path, some participants failed to provide correct answers to this question, especially at lower transparency levels. Q3 asks what is constraining the generated path. The constraint implemented in the measurement phase was the maximum allowable path length. The results of Q3 suggest that the constraint should be more clearly indicated in the visualisation. For example, a text box displaying “current path length / maximum allowable path length” can be attached.

Q4 (discretisation) and Q8 (overall) pertain to the advantages and disadvantages of the algorithms, which are difficult to answer because they cannot be directly perceived and need to be inferred from the visualisation. Q5 is related to search nodes. There is an option in Q5 that misled many participants: the search node represents a visible region from a certain location. This option was designed originally for Anya and Polyanya. The search nodes of Theta* and Informed RRT* are not regions but points, making this option



(a) Theta*



(b) Informed RRT*

Figure 3.7: The number of correct answers for each question at each transparency level.

incorrect for both algorithms. However, the visualisation cannot present points without a radius. For Theta*, the search nodes are represented by grid cells rather than points located at the grid centres. This misled participants, especially at Levels 4 and 5, causing them to change their correct answers to incorrect ones.

Q6 is related to search trees. The visualisation of search trees (Level 4) positively impacts the accuracy of responses to Q6. Q7 asks about the strategy of the path-planning algorithm to find the shortest path (straighten the search tree). The results indicate that this question is difficult to answer. Some participants suggested that at Level 5, the processes of expansion, rewiring, and pruning should all be presented, rather than just the final search tree of each step. Visualising these intermediate sub-procedures at each step could help humans better understand the search process and strategy.

For further analysis, statistical tests were performed. Mann-Whitney U tests found a significant difference in the hit ratio between Theta* and Informed RRT* ($W = 94.5, p = 0.004, r = 0.707$). This result suggests that Informed RRT* may be easier to comprehend than Theta* via visualisation, which contradicts the initial hypothesis (H2). One possible reason is that, although the random sampling of Informed RRT* makes its behaviour difficult to predict, this randomness also makes its strategy and inner workings easy to observe. For example, by viewing the random sampling points generated by Informed RRT* (Level 2), participants may gain some insights into how the algorithm explores the space to find paths. In contrast, Theta* generates the same result given the same input. At lower levels of transparency, a few participants believed that the search tree of Theta* does not change during the algorithm's execution (Q6), and some participants assumed that Theta* considers the vertices of obstacles because it generates paths that appear as straight as possible (Q5 and Q7). These results suggest that the algorithm type may influence human understanding of path-planning algorithms. For Theta*, a deeper level of transparency may be more necessary compared to Informed RRT* to help humans form a more accurate mental model.

Friedman tests found significant differences in the hit ratio between the transparency levels in both Theta* ($\chi^2(4) = 25.085, p < 0.001, w = 0.314$) and Informed RRT* ($\chi^2(4) = 16.446, p = 0.002, w = 0.206$). The effect sizes are not large, indicating that the substantive differences between transparency levels are not particularly pronounced. This is potentially because the number of questions is limited and some questions are too difficult to answer (e.g., Q3, Q5 and Q7 in Figure 3.7). Participants generally gained low hit ratios at each level (see Table 3.1). Moreover, the information was cumulatively presented as the level of transparency increased. The magnitude of differences between adjacent transparency levels is indeed not very large in terms of information amount (except for Level 5, search process). In Theta*, exact pairwise comparisons with Bonferroni correction further revealed significant differences in the hit ratio between Level 5 and Level 1 ($D = 41.5, p < 0.001$), Level 4 and Level 1 ($D = 30.0, p = 0.028$). In Informed RRT*, exact pairwise comparisons with Bonferroni correction further revealed a significant difference in the hit ratio between Level 5 and Level 1 ($D = 32, p = 0.013$). Compared to Level 1 (baseline), Level 4 (search tree) and Level 5 (search process) could offer significant insights into algorithms.

3.5.3. CONFIDENCE

Figure 3.8 depicts the distribution of confidence in being unable to answer, answering correctly, and answering incorrectly at each transparency level. Overall, the confidence tends to increase with the transparency level, regardless of whether the answer is correct or not. This observation is in line with other research on transparency [25] and decision-making [165]. Although the self-rated confidence is related to the amount of information received, the confidence in correct answers is generally higher than in incorrect answers, which also aligns with the findings of Daun et al. [166]. This indicates that the participants' self-rated confidence can indeed reflect their understanding to a certain degree. The confidence in being unable to answer decreases with increasing transparency, possi-

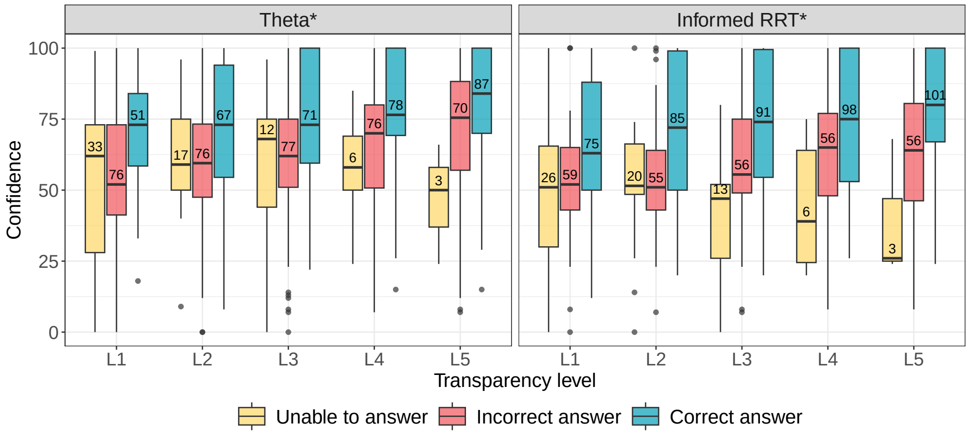


Figure 3.8: Changes in confidence with increasing levels of transparency. The number above the median line indicates the number of observations in each box plot.

Table 3.2: Exact pairwise comparisons of confidence between transparency levels (with Bonferroni correction).

	Theta*					**Informed RRT***				
	Level 1	Level 2	Level 3	Level 4	Level 5	Level 1	Level 2	Level 3	Level 4	Level 5
Level 1										
Level 2	1.000					0.308				
Level 3	0.990	1.000				0.028*	1.000			
Level 4	**	0.028*	0.308			**	0.028*	0.308		
Level 5	**	**	**	0.237		**	**	0.019*	1.000	

Note: $^{*}p < 0.05$, $^{**}p < 0.01$.

bly because participants felt they should be able to answer as the amount of information increases, even if they still failed.

Friedman tests revealed significant differences in the confidence between the transparency levels in both Theta* ($\chi^2(4) = 50.394, p < 0.001, w = 0.630$) and Informed RRT* ($\chi^2(4) = 47.251, p < 0.001, w = 0.591$). In contrast to the hit ratio, the substantive differences in the confidence are evident. This is likely because confidence is independent of the number of questions and participants more easily became confident as they received more information [165]. Exact pairwise comparisons with Bonferroni correction further revealed differences between transparency levels, as shown in Table 3.2. The results indicate that Level 4 and Level 5 could significantly improve perceived understanding.

3.5.4. CALIBRATED UNDERSTANDING

Figure 3.9 illustrates the calibrated understanding based on the hit ratio and confidence, which reflect the tested and perceived understanding, respectively. The "unable to answer" responses were counted as incorrect answers with zero confidence in this figure, regardless of the original confidence ratings given by participants. This is because, when

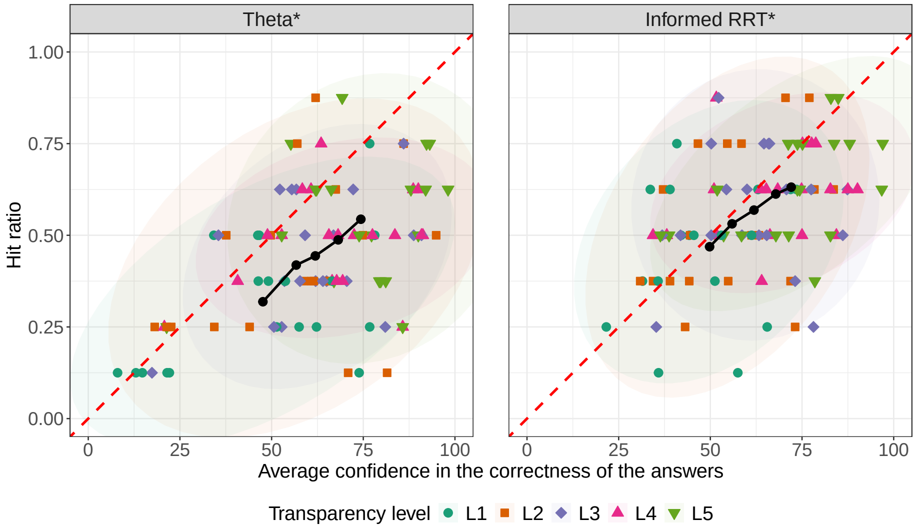


Figure 3.9: Calibrated understanding by hit ratio and confidence, featuring 95% confidence ellipses and their centres (black dots) for each transparency level.

participants selected this option, they already knew their answers were incorrect (zero confidence in correctness). In contrast, when respondents selected other options, their ratings indicated a higher degree of confidence in the correctness of their answers.

It was expected that a well-calibrated understanding would align with the diagonal line (from bottom left to top right) as the transparency level increases [157]. Figure 3.9 indicates that despite high variability in calibrated understanding among participants, the desired trend is generally evident in the mean (black lines). Compared to Informed RRT*, participants may exhibit overconfidence (i.e., a low hit ratio with high confidence) when learning about Theta*. This indicates that the algorithm type may also play a vital role in human understanding.

3.5.5. LEARNING TIME

Figure 3.10 presents the learning time distribution of Theta* and Informed RRT* at five transparency levels for the eight different questions. Overall, Theta* demonstrates lower learning time compared to Informed RRT*. As transparency levels increase, a downtrend in learning time is observed for both algorithms, suggesting that higher transparency may facilitate more efficient learning. However, the extent of this change varies by question. For Q7, Informed RRT* takes more time at Levels 4 and 5. This is likely because Levels 4 and 5 enable participants to observe the strategy of Informed RRT* in detail, but the processes of node expansion and tree rewiring are not easy to understand. For Q3, the constraint was not clearly stated, causing participants to spend much time exploring the algorithms to determine the exact parameter that was constraining the algorithms.

Mann-Whitney U tests found no significant difference between Theta* and Informed RRT* in terms of the total learning time. However, Table 3.1 suggests that participants

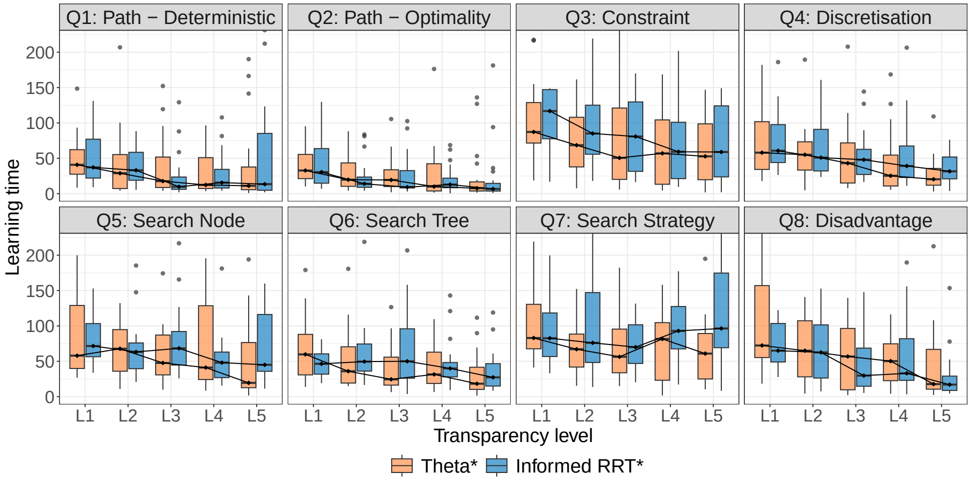


Figure 3.10: Learning time for each question at each transparency level.

may learn Theta* more quickly than Informed RRT*. This is probably because Theta* is a deterministic, grid-based algorithm. It always generates the same result under the same conditions in the same environment and its graph (i.e., grid) remains fixed regardless of changes in obstacles. Each time a participant added or removed an obstacle, the visual information of Theta* did not change much on the display. This may enable participants to become familiar with Theta* more quickly.

Friedman tests found significant differences in the learning time between the transparency levels in Theta* ($\chi^2(4) = 21.72, p < 0.001, w = 0.272$), but no significant difference in Informed RRT*. In Theta*, similar to the hit ratio, the effect size for the learning time is small. This may also be attributed to the correlations between transparency levels, which allowed participants to quickly grasp the newly added information. Moreover, the experiment had a maximum time limit, requiring participants to effectively manage their time for each level. Exact pairwise comparisons with Bonferroni correction further revealed significant differences in the learning time between Level 5 and Level 1 ($D = -43, p < 0.001$), Level 4 and Level 1 ($D = -36, p = 0.003$). Participants tended to spend more time on Level 1 because it was their first experience with the new algorithm. Participants spent more time checking Level 5 of Informed RRT* because understanding the random expansion of nodes and the rewiring of search trees is generally difficult.

3.5.6. INTERACTION

There are two metrics for measuring interactions: the number of algorithm executions and the number of interactions with the map. Algorithm executions represent how many times participants challenged the algorithm by adding/removing obstacles that would lead to observable changes to their results, while interactions with the map mainly refer to the number of obstacles added or removed. Given that algorithm executions reflect the frequency with which participants see new information, this metric is used for analysis.

Figure 3.11 shows the number of algorithm executions of Theta* and Informed RRT*

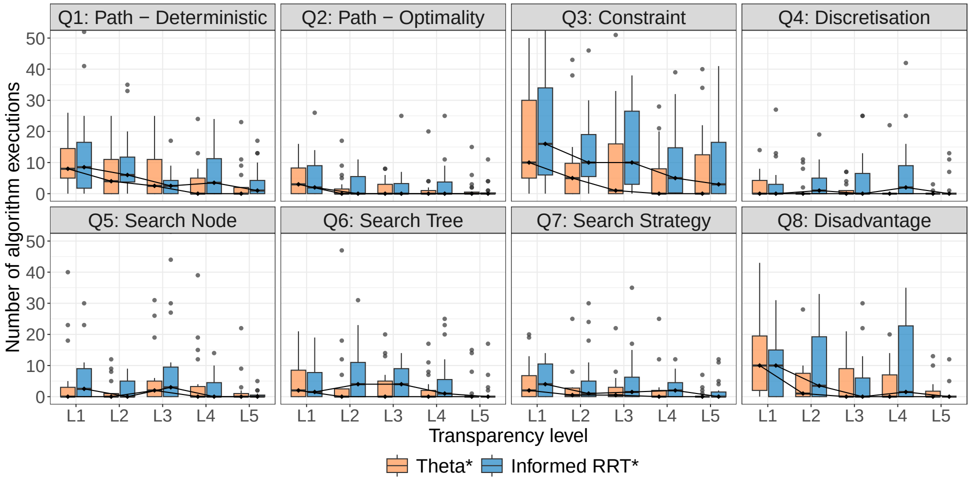


Figure 3.11: The number of algorithm executions (reroutes) for each question at each transparency level.

for each question at each transparency level. The trend of this metric is similar to that of the learning time (see Figure 3.10). In the figure, Q3 is the most prominent, as participants spent much effort creating various scenarios to examine the constraint. Informed RRT* generally exhibits higher median execution numbers and greater variability compared to Theta*. This is because Informed RRT* employs a random sampling strategy and some participants frequently executed Informed RRT* in the same scenario and observed changes in the results. Mann-Whitney U tests found no significant difference between Theta* and Informed RRT* in terms of the number of algorithm executions.

Friedman tests found significant differences in the number of algorithm executions between the transparency levels in both Theta* ($\chi^2(4) = 21.7, p < 0.001, w = 0.271$) and Informed RRT* ($\chi^2(4) = 19.719, p < 0.001, w = 0.246$). The effect sizes are both small, indicating that while differences in the interactions between transparency levels exist, they are not apparent. In addition to the relatively small differences between adjacent transparency levels, the fixed interaction mode and starting and target points (control variables) also reduce the need for a large number of interactions to achieve understanding.

In Theta*, exact pairwise comparisons with Bonferroni correction further found significant differences in the algorithm executions between Level 5 and Level 1 ($D = -42, p < 0.001$), Level 4 and Level 1 ($D = -32.5, p = 0.009$). In Informed RRT*, exact pairwise comparisons with Bonferroni correction further revealed significant differences in the algorithm executions between Level 5 and Level 1 ($D = -42.5, p < 0.001$), Level 5 and Level 2 ($D = -28.5, p = 0.039$), Level 5 and Level 3 ($D = -28.5, p = 0.039$), Level 5 and Level 4 ($D = -30.5, p = 0.019$).

This result can be linked to the learning time. The Spearman's rank correlation coefficient (0.625) indicates a strong correlation between the learning time and the number of algorithm executions. It seems that the greater the number of interactions, the longer the learning process takes. At Level 5 of Informed RRT*, the algorithm was executed less frequently, but the learning time was longer because participants spent considerable time examining the search process.

3.5.7. PREFERENCE

Figure 3.12 illustrates the Likert scale ratings for the transparency elements. The expansion, rewiring and pruning processes are integrated into the search process. Generally, except for the cost values, all other elements are considered highly useful for understanding both Theta* and Informed RRT*. This is probably because the tested algorithms both aim to find the shortest path and the cost value is intuitively reflected in the path length. However, for other types of costs in more real-life operational contexts, such as fuel consumption and risks, the cost values and their weights may become more useful. Compared to Informed RRT*, more participants found the search space and graph elements less useful for understanding Theta*. This may be because these elements can better help participants understand the random sampling strategy of Informed RRT*. However, for Theta*, they appear redundant due to the explored nodes and search trees.

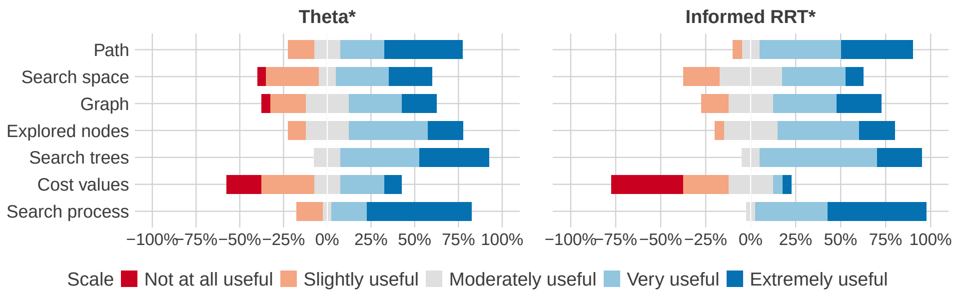


Figure 3.12: Likert scale ratings for the transparency elements regarding the usefulness for understanding the path-planning algorithms.

3.6. DISCUSSION

3.6.1. SAMPLING-BASED ALGORITHMS VS. GRAPH-BASED ALGORITHMS

According to the experiment results, it seems that sampling-based path-planning algorithms are more suitable for applications requiring algorithmic transparency, as participants were generally able to better understand sampling-based algorithms than graph-based algorithms.

Before conducting the experiment, it was hypothesised that Informed RRT* would be more difficult to understand than Theta* (H2), due to its random sampling strategy and less organised visual presentation. However, the results indicate the opposite. This may be because the random sampling for path planning is more meaningful to humans than expected. Each time the algorithm was executed, Informed RRT* generated different random points, making it salient to participants that the algorithm used a random strategy to explore the space. For Theta*, although it is deterministic and highly structured, some participants mistakenly thought it also used a random strategy for exploration because its current node appeared to jump around randomly at each step. They did not understand why the current node behaved this way (recall that the current node is the node with the lowest f-value in the *open* list in Section 2.2). For better understanding, it may be necessary to more clearly indicate the basis of the current node selection in the visualisation. For example, provide additional textual and/or verbal explanations to clarify the rule, rather than requiring users to infer it from the visualisation.

While the sampling-based algorithms showcased several advantages in this research related to understandability, they cannot replace graph-based algorithms. The main disadvantage of sampling-based algorithms is that they do not fully utilise the prior knowledge of maps and they may sometimes get stuck in narrow passages [101, 102]. Stability and optimality are still the biggest obstacles for sampling-based algorithms in practice. Rather than drawing a rash conclusion ("which one is better"), this research offers a new perspective for comparing algorithms on transparency and understanding criteria. These dimensions are also important to consider in real-world applications, especially in high-stakes domains where technology requires human oversight.

3

3.6.2. IS INCREASED TRANSPARENCY ALWAYS BETTER?

The user study reveals that increased transparency generally leads to better understanding, as hypothesised (H1). This result aligns with findings from other research [30, 142].

In the experiment, participants tended to rely on their expectations to formulate answers, especially when lacking sufficient information. The increased transparency can continuously correct their misconceptions and refine their mental models of the algorithm. However, this does not mean that they only changed incorrect answers to correct ones. Sometimes, when new information at subsequent levels violated participants' expectations, they felt more confused and provided even more incorrect answers, resulting in a lower hit ratio. This phenomenon is similar to the mixed results identified in other research [30, 138], suggesting that more information does not necessarily lead to better understanding and may instead result in greater confusion. This may occur when new information is insufficient to alleviate their confusion caused by the mismatch between their expectations and the truth, or the new information is unclear.

For example, Theta* is an any-angle path-planning algorithm that ignores the edges of the graph (e.g., eight adjacent grids). The mismatch between the paths and the graph (Level 2) made participants feel more confused. At lower levels of transparency, they initially thought that Theta* primarily explored the vertices of obstacles rather than grid by grid. This confusion also occurred when participants noticed that some search nodes and sampling points in Informed RRT* did not overlap. This inconsistency between the truth and human expectations indicates that higher levels of transparency are required to deepen understanding for some algorithms. It has been observed that participants' confusion was indeed alleviated (more confident and higher hit ratio) with higher levels of transparency. When the information is clear to users, they should eventually be able to gain a deeper understanding.

In the experiment, it was found that participants' confidence increased with higher levels of transparency, regardless of whether their answers were correct or incorrect. This indicates that participants can become overconfident as transparency increases. This may be because participants misunderstood some transparency information, believing they understood it correctly when they actually did not. To alleviate this problem, one possible solution is to further increase transparency and provide more detailed, clear information. As learning deepens, users may identify their own mistakes. Another possible solution is to evaluate users' understanding and provide feedback to promptly correct their mental models. This idea is similar to the concept of model reconciliation [74] as mentioned in the introduction, but estimating human mental models in real time still remains a challenging issue.

Fortunately, the procedures and strategies of algorithms are usually fixed, allowing users to learn them during training. In real-time operations, users may pay more attention to situational changes. If users already grasp the algorithm's underlying principles, transparency information could help them quickly understand the algorithm behaviour in a certain scenario, thereby improving their situation awareness [70, 135].

The statistical analyses found significant differences in the dependent measures between transparency levels, but mostly with small effect sizes. This is largely because of the dependent relationships between transparency levels, where higher levels are built upon lower ones (see Figure 3.2). As the level of transparency increases, the information regarding the inner workings of the algorithm is progressively and cumulatively revealed. If the information were chunked into more levels, the differences between transparency levels might be even less obvious.

The results suggest that perhaps certain levels of transparency can be combined or removed to obtain potentially large effect sizes. For example, based on the benchmark tests in Chapter 2 and the user study findings, Level 4 (search tree) strikes a good balance between algorithm speed, human understanding and visual presentation complexity. Compared to Level 4, Levels 2 and 3 might be redundant and could be removed. Meanwhile, Levels 1 and 5 should be retained considering that Level 1 pertains to domain constraints (distinct type of information) and Level 5 to the search process (animation that contains a large amount of information). The elements "search space", "graph" and "explored nodes" can also be separated from Level 4 to further condense the transparency information.

3.6.3. THE ROLE OF ENGAGEMENT IN UNDERSTANDING AND OPERATIONS

As indicated by Hundhausen *et al.* [40], Naps *et al.* [127], Grissom *et al.* [152], Doherty and Doherty [167], engagement plays a crucial role in enhancing human understanding. Although this research did not specifically assess the impact of engagement, efforts were made to maximise the participant engagement during the experiment (control variables). According to the engagement taxonomy in CS education [127], there are six categories for engagement with algorithm visualisation: 1) No Viewing, 2) Viewing, 3) Responding, 4) Changing, 5) Constructing, and 6) Presenting. "No Viewing" refers to instruction without any visualisation technology whereas "Viewing" is the core form of the visualisation engagement. "Responding" involves answering questions concerning the visualisation and "Changing" allows learners to change the algorithm's input to explore its behaviour. "Constructing" indicates learners build their own algorithm visualisations and "Presenting" suggests showing a visualisation to an audience for feedback and discussion. This research mainly incorporated "Viewing", "Responding" and "Changing" categories. Since the target audience was primarily non-experts, "Constructing" and "Presenting" were deemed less suitable.

For real-time operations, operators perform more effectively when their engagement level is high [70, 168]. As the level of automation increases, operators still need to be actively engaged into operational tasks and the control loop to understand what the system is doing [129, 169]. Otherwise, they could lose situation awareness and be unable to detect and address the system errors promptly. Empirical evidence has found a positive effect of transparency on situation awareness and operator performance due to the increased information [19, 24, 136]. In fact, transparency can also increase the engagement level of operators by offering ways to interact with the system. For example, transparency can allow operators to visually examine the inner workings of path-planning algorithms, helping them understand why and how a specific path was proposed. Additionally, it can assist operators in conducting "what-if" analyses [170]. Operators may actively check the transparency information, thus avoiding passively monitoring the system for extended periods and becoming bored.

3.6.4. LIMITATIONS AND FUTURE DIRECTIONS

The user study suggests that the designed visualisation may need further improvement. For example, detailed information related to the search space and constraints should be provided, such as "current path length / maximum allowable path length". This would make it easier for users to check the relationship between the current results and the constraint bounds. For operational purposes, it is better to state the constraints in advance rather than letting users guess. In grid-based algorithms, if the search node is generated at the centre of the grid rather than at the vertex, it may be helpful to mark this centre point instead of the entire grid (or explain in advance). Marking the entire grid is a common approach to visualise the search node for grid-based algorithms, but it may mislead users into thinking that the algorithms explore the entire region within that grid, while actually only the centre will be explored. Furthermore, the search tree is portrayed at each step to reveal the search process (Level 5). However, the search process actually includes expansion, rewiring, and pruning, which can be separated for clearer presentation.

In addition to the visualisation, textual and verbal explanations can also be useful in some cases. Textual explanations excel in providing precise, detailed descriptions that may be hard to capture in a visual format, such as abstract concepts and numerical data. In contrast, verbal explanations are inherently more human-like, fostering an interactive environment and deeper engagement for users. This research does not compare visual transparency with textual or verbal transparency because there are no readily available textual or verbal explanations for how path-planning algorithms work. To create them, it would be required to extract explanations from the original papers that introduce these algorithms, such as Theta* [66] and Informed RRT* [83]. Moreover, determining the appropriate level of detail to include in the textual or verbal explanations would also need to be addressed, which is beyond the scope of this study. For future research on transparent path planning, this work could serve as a reference and a basis for comparison. The differences in effectiveness and user preferences between visual, textual, and verbal transparency could be further explored [145].

The user study assessed the participants' understanding mainly through the eight questions, which were designed only to evaluate the understanding of the algorithm's inner workings. Future research could incorporate more questions to assess broader aspects of transparency and understanding, such as debugging, potential algorithm improvements and the environmental impact of solutions. Many participants reported that some questions were sometimes difficult to answer, which explains the relatively low average hit ratio. This may be because most participants were novices or beginners in path planning and were not familiar with its basic concepts. They needed to learn the relevant terms, quiz questions and options, and, most importantly, the algorithm itself within a limited time (20 minutes for training and 60 minutes for measurement). This task was non-trivial for many of them.

Additionally, the user study was intentionally designed in a generic setting, as path-planning algorithms generally follow fixed procedures, and revealing their inner workings does not require a specific real-world context. However, domain experts with extensive operational background may have better initial understanding of algorithms within their domain [150]. Future studies should therefore also consider how domain knowledge and operational expertise and experience may impact (the need for) transparency in understanding the working mechanisms of algorithms.

Finally, the proposed transparency levels were implemented to be adaptable in the web-based pathfinding visualiser, enabling users to drive the interaction with the interface and access information on demand. This approach could avoid "automation surprises" from unexpected autonomy-driven changes in real-time operations [171]. It is particularly suitable for human-centred domains such as air traffic control, where operators should maintain authority over and awareness of automation. Adaptive transparency that considers users' cognitive load and expertise levels could also be a promising area to explore. For example, by detecting users' confusion emotion [172], the system could automatically adjust the level of transparency to provide more or less information and explanations. The ultimate goal of this research is to enhance human understanding through algorithmic transparency, rather than dynamically tailoring information delivery based on users' real-time cognitive states. The exploration of adaptive transparency is beyond the scope of this study. All in all, whether transparency is adaptive or adaptable, the findings of this research could provide a reference for both.

3.7. CONCLUSION

This chapter extends the work presented in Chapter 2 and proposes six levels of visual transparency for path-planning algorithms. A user study was conducted to explore the impact of different transparency levels on human understanding. Results demonstrate that visual transparency allows inexperienced people to correctly and confidently understand the details of a path-planning algorithm and increased transparency generally improves understanding. The user study also shows that a lack of prior experience and knowledge of path-planning sometimes can lead to confusion at higher levels of transparency, for various reasons but especially when the algorithm behaves contrary to expectations. Training, combined with a clear operational context and an algorithm that matches human expectations, can probably alleviate the need for greater transparency.

4

TOWARDS A UNIFIED TRANSPARENCY TAXONOMY

In the previous chapters, transparent path planning and its impact on human understanding were explored, without focusing on a specific operational context. This chapter further extends the research scope to tactical UTM scenarios (e.g., drone rerouting), aiming to achieve transparent UTM. A unified transparency taxonomy is developed, integrating operational, domain, and engineering transparency. A survey study is performed to investigate the transparency needs of operators for supervising UTM and validate the effectiveness of the proposed taxonomy.

The contents of this chapter are based on:

Paper title	Towards a unified taxonomy for algorithmic transparency: Insights from uncrewed air traffic management
Authors	Yiyuan Zou and Clark Borst
Published in	Cognition, Technology & Work
DOI	10.1007/s10111-025-00826-5

ABSTRACT

With the rapid advancement of drone technology, drone applications have become increasingly widespread. However, the integration of drones into the airspace also poses risks to crewed aircraft, particularly around airports. To address this issue, highly automated Uncrewed Air Traffic Management (UTM) systems are being developed worldwide. Since fully safe and reliable automation does not exist yet, UTM still requires human supervision to enhance the overall system safety and reliability. Some form of "seeing-into" transparency may be necessary to help operators better understand the limitations and behaviour of the automated UTM system. As UTM is a novel concept, research on transparent UTM is limited. Many efforts have been made in other fields, but there still remains a lack of consensus on what transparency entails, particularly for algorithmic systems. Therefore, this chapter first presents a unified taxonomy for algorithmic transparency, with operational, domain and engineering transparency introduced as its core concepts. From the taxonomy, twenty UTM transparency elements and their corresponding visual prototypes were then designed, which also showcases how the taxonomy can be applied in practice. A survey-based user study was conducted to collect the opinions of air traffic controllers and drone experts regarding the designed elements and prototypes. Results indicate that transparency is deemed imperative for UTM, especially in scenarios featuring automation failure. It also reveals that operational transparency is generally preferable over engineering transparency in nominal operations. Participants were asked to group the designed elements, and their results closely aligned with the structure of the proposed taxonomy.

4

4.1. INTRODUCTION

In recent years, drone usage has rapidly increased in various domains, such as agriculture, inspection, delivery, surveillance and entertainment. In aviation, it is expected that a large number of drones will share the airspace with crewed aircraft in the near future [2, 3]. To safely cope with the increased number of drones, Uncrewed Air Traffic Management (UTM) was proposed and is currently under development [6, 8, 9, 173, 174]. As UTM is a novel concept, a universally recognised standard has not yet been firmly established [175]. Various solutions for UTM are being actively explored around the world, such as American UTM [174], European U-space [9] and Chinese UTMISS [1].

Despite the differences among these solutions, there is a consensus that, unlike traditional Air Traffic Management (ATM), UTM will be built from the ground up to rely on high levels of automation. This is because drone traffic often involves much higher flight densities and could be far more complex than existing crewed air traffic. It is nearly impossible for humans to manually control such a large number of drones simultaneously. However, relying heavily on automation may also be problematic, especially in the low-altitude airspace around airports where drones and crewed aircraft coexist. Any deficiencies or limitations in automation could increase the risk of collisions between drones and crewed aircraft, posing a threat to human lives. Therefore, although UTM is highly automated, it still requires human supervision, at least in the Controlled Traffic Regions (CTR) around airports, to enhance the overall safety and reliability.

However, a higher level of automation generally makes it more difficult for human supervision [70, 169]. Operators may be unable to understand the automation decisions and their underlying reasoning without additional support. In this case, they are left no choice but to either blindly follow the actions suggested by automation or manually control everything due to a lack of trust in automation. In UTM, if a drone's behaviour is completely unpredictable and uninterpretable, it will pose a huge threat to crewed aircraft and should not be allowed to operate around airports [4]. To address this issue, some form of "seeing-into" transparency may be needed that presents information and/or explanations about the inputs, outputs and internal processes of automation [19, 20, 136, 176]. UTM operators should be aware of the intents and goals of both drones (e.g., where are their destinations?) and the automated UTM system (e.g., what automated services are provided?). Safety-related metrics, such as Closest Point of Approach (CPA), could also be disclosed to help operators monitor the situation [55].

Since UTM is not fully developed, direct research on transparent UTM is limited [14–16, 59, 63, 177]. In other related fields, some studies have been conducted to explore the design and impact of transparency [19, 33, 136], such as the Situation Awareness–based Agent Transparency (SAT) model [22, 134], Ecological Interface Design (EID) [53, 56] and Explainable AI (XAI) design frameworks [114, 131]. Previous research also suggested positive effects of transparency on human performance in one-to-many drone operations [178] and multi-unmanned (air, ground, and sea) vehicle mission planning [23–25]. However, these works generally stem from different perspectives on transparency, and there remains a lack of consensus on what transparency entails, particularly for algorithmic systems. In the UTM context, it is still unclear how to achieve transparency and how transparency could affect the collaborative operations between drones and crewed aircraft. Therefore, the goal of this chapter is to propose a unified taxonomy for algorithmic transparency to integrate different perspectives. Based on this taxonomy, various information elements and visual prototypes can be devised for transparent UTM. According to the designed elements and prototypes, a survey-based user study can be conducted to validate the proposed taxonomy and explore operators' needs and preferences for transparency in different UTM scenarios.

This chapter is organised as follows. Section 4.2 reviews different perspectives and related works on transparency and presents a unified transparency taxonomy. Based on the taxonomy, Section 4.3 shows the design of twenty transparency elements and fourteen corresponding visual prototypes for UTM. Section 4.4 outlines the user study methodology, introducing the questionnaire structure and participant background. Section 4.5 presents the results collected from participants, analysing their needs and preferences regarding the designed transparency elements and prototypes. Section 4.6 discusses the insights gained from the user study and future research directions.

4.2. TRANSPARENCY TAXONOMY

Much research has been conducted to enhance automation transparency [19, 31, 33, 136, 179], but their methods are usually diverse and lack a unified guide or framework. For example, the Single European Sky ATM Research (SESAR) projects ARTIMATION [60], MAHALO [61] and TAPAS [55] all explored methods to achieve transparency in tactical ATM operations, yet ended up with different design choices. Some studies [23, 25, 55] emphasised improving operator situation awareness in human-automation collaboration through transparency. They usually followed Endsley's situation awareness theory [135] and Chen's Situation Awareness–based Agent Transparency (SAT) model [22, 134]. In contrast, other studies [58, 61, 107] focused on revealing the physical and intentional constraints of work domains to establish a common ground for both humans and machines. This approach was mainly based on Ecological Interface Design (EID) [53] and Cognitive Work Analysis (CWA) [57]. Additionally, some other research [60, 180, 181] seemed to draw inspiration from the Explainable AI (XAI) community, aiming to design explainable models to elucidate the inner workings of automation.

Although transparency can be approached from different perspectives, the overall goal is to support human supervision of automation and ensure humans remain in the loop. The ecological approach is developed for improving operator situation awareness as well [54], while the XAI methods are implemented to foster the trust and acceptance of human users [181]. Essentially, transparency is about disclosing relevant information to humans, and thus the focus of transparency may shift depending on the needs and background of the user [111]. Operators may be more concerned with how automation affects operational scenarios and situations, whereas policymakers may need to assess whether an automated system is trustworthy and reliable for real-world application. Different needs may lead to different methods and perspectives on transparency.

However, even for the same type of users, their transparency needs may still be different depending on personal preferences, expertise, and specific contexts [115, 134]. In addition to situation-related information, operators may also seek to understand the inner workings of automation, especially when automation behaves unexpectedly [14]. To meet user needs across all scenarios, it seems necessary to implement an overarching approach to disclose as much information as possible. Users can thus access the information they need on demand. Therefore, a unified *transparency taxonomy* is devised to summarise different perspectives on transparency. The proposed taxonomy could also serve as a guide for transparency design.

As this research focuses on tactical UTM operations within CTR around airports, it primarily investigates the needs and preferences of operational users, such as Air Traffic Controllers (ATCos) and drone experts, rather than those of policymakers or system developers/designers in the survey-based user study.

4.2.1. PERSPECTIVES ON TRANSPARENCY

Three main perspectives can be summarised from the literature: *user-centred* [21, 22, 30, 131], *model-centred* [71, 94, 95] and *ecology-centred* [53, 107, 108].

From a user-centred perspective, transparency information should be presented in accordance with user demands, limitations, preferences and expertise [182]. For example, Lyons' human-robot transparency model [21] defines various types of information that need be presented to humans, such as robots' tasks, purposes, decision-making processes and environmental perceptions. To prevent overwhelming users, transparency is generally divided into different levels, enabling a progressive and incremental disclosure of information [30]. For example, corresponding to the three levels of situation awareness defined by Endsley [135], Chen et al. [22] designed three levels of transparency in their SAT model: Basic Information (Level 1), Rationale (Level 2) and Outcomes (Level 3). In practice, these transparency levels are usually combined in visual and/or textual presentations [24]. However, information revealed by the SAT model might be insufficient in some cases. When automation fails or behaves unexpectedly, users may seek more information about the agent's internal process (i.e., *how* the agent makes decisions) to understand what happened, why it happened, and how to resolve it [76, 77]. This type of information is not explicitly reflected in the SAT model.

The model-centred approaches are mostly developed in the XAI community, aiming to construct explainable models that are readily comprehensible to humans. As technology advances, automation becomes increasingly complex and difficult for humans to understand, such as neural networks and reinforcement learning. XAI was thus introduced to enhance the explainability of advanced AI models and algorithms [31]. Many approaches have been developed, including Local Interpretable Model-agnostic Explanations (LIME) [93] and Shapley Additive Explanations (SHAP) [94]. The main focus of the model-centred perspective is to thoroughly dissect the internal processes of models and explain them in human-understandable terms. In recent years, user-centric XAI has gained increasing attention from researchers since explanations should also consider user expertise and needs [33]. For example, a theory-driven user-centric framework for XAI has been proposed [131], aiming to support human reasoning and mitigate cognitive biases through tailored explanations.

XAI has also been applied in ATM to improve the operator trust and acceptance of AI-based ATM systems [180, 181, 183]. As the core of UTM is to guide drones to their destinations while avoiding conflicts with crewed aircraft, a centralised conflict-free path-planning algorithm is expected to be implemented to achieve this goal [63]. Therefore, this study primarily focuses on transparent path planning. Many pathfinding visualisers have been developed to portray the search processes of various path-planning algorithms [43, 44, 48, 100], which could also serve as references for this work.

The ecology-centred approach is derived from EID [53, 153] and CWA [57]. It puts emphasis on visualising the physical and intentional constraints governing the work domain, revealing its deep structure for achieving *domain* transparency [107, 108]. This approach can provide a common ground for user-centred and model-centred approaches since both humans and machines should obey the same domain constraints. Technically, the ecological approach seeks to discover the most effective way for presenting domain constraints, fully utilising the human ability for direct perception [184, 185]. For instance, in drone flight monitoring, drone endurance can be depicted as a virtual battery [186] or alternatively represented as available manoeuvring space (an elliptical space) [14]. The latter may be more intuitive for humans in the context of path planning since it builds a direct link between the constraint and the solution: the path should be within the manoeuvring space to satisfy the endurance constraint.

The ecology-centred approach could provide users with deeper insights into the *solution space* of a task, fostering a clearer understanding of the feasibility and robustness of solutions as well as serving as input/output feature spaces for human intervention. The ecology-centred approach has yielded many promising results for ATM and aviation [54, 107, 187–189]. In essence, human and automated agents alike are constrained by the same fundamental laws and causalities that govern the work domain. Contextualising machine intentions can enhance the comprehension of their underlying motives.

While these three perspectives differ from one another, they are connected when considering a triadic semiotic perspective on socio-technical systems [190], as depicted in

Figure 4.1. The model-centred design identifies what content about machines can be revealed, including implemented algorithms, internal processes, and reasoning mechanisms. The user-centred design determines how the information should be conveyed to humans, taking the goals, skills and preferences of human users into account. The ecology-centred design highlights domain constraints, such as laws of physics, principles and dynamics. The behaviour of both humans and machines should be bounded by these constraints. The interface acts as a bridge, facilitating interactions and communications between humans and machines. The transparency information is usually presented on an interface, conveying the state of and constraints within the environment, what the machine does, and what the users may need and/or prefer.

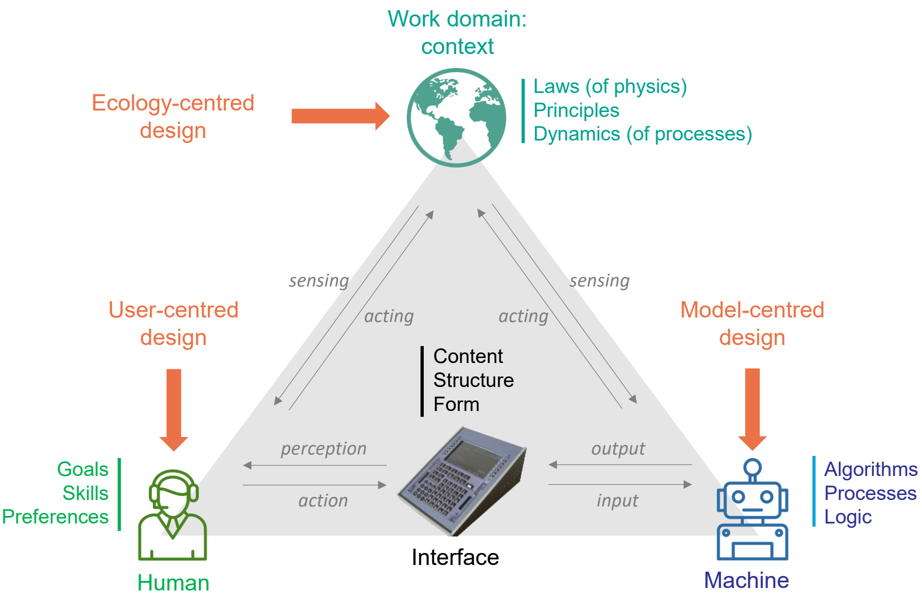


Figure 4.1: Triadic perspective on transparency, capturing user-, model-, and ecology-centred perspectives.

4.2.2. TRANSPARENCY IN ATM AND UTM

In the fields of ATM and UTM, automation transparency is gaining increasing attention. SESAR 3 Joint Undertaking initiated five projects [36] to address transparency issues of AI in ATM. The most relevant projects for tactical UTM operations are ARTIMATION [60], MAHALO [61] and TAPAS [55].

ARTIMATION proposed three levels of transparency: 1) Black Box, 2) Heat Map and 3) Storytelling. The Black Box showed only the proposed solution along with the instructions for execution. The Heat Map presented what trajectory was explored by the algorithm and whether it was good or bad. The Storytelling provided a step-by-step preview of the proposed solution while also explaining alternative possibilities. A user study of ARTIMATION [60] showed that ATCos preferred the Black Box because of the time pressure issue. They favoured the simplest interface design for Air Traffic Control (ATC). They also thought that transparency would be beneficial for the initial training period to increase the understanding and trust of ATCos. In real operations, transparency should be hidden by default but remain accessible.

MAHALO devised three transparency conditions: 1) Vector Line, 2) Vector Line and Solution Space Diagram (SSD) and 3) Vector Line, SSD and text-based explanation. The Vector Line, indicating flight speed and heading, represented the proposed solution for conflict resolution. The text-based explanation clarified the target Closest Point of Approach (CPA) and the agent's purpose. The core of MAHALO is the SSD, which could *visually* explain whether the proposed solution is feasible and how robust it is. MAHALO also explored the personalisation of AI to align its advice more closely with ATCos' preferences. A user study of MAHALO [61] indicated that personalised advisories were more easily accepted by ATCos than transparent advisories and that greater personalisation may reduce the need for transparency.

TAPAS did not have explicit transparency levels in their Conflict Detection & Resolution (CD&R) use case. Instead, it mainly utilised text-based tables to present detailed information and possible solutions associated with CD&R, such as geometrical features of the CPA detected and suggested actions along with their expected outcomes. Their transparency design is similar to Chen's SAT model, which is also based on Endsley's situation awareness theory [135]. A user study of TAPAS [55] showed that providing information that maintains operators' situation awareness could probably be sufficient to develop trust in AI, even in high-stakes fields like ATC.

To summarise, these projects all had different perspectives on addressing the same problem (i.e., CD&R in ATC). Each of them developed its unique transparency elements, covering different visual and textual parameters representing the tactical ATM context. Nonetheless, some similarities were found in that they all centre transparency information around *solutions*, revealing information about the proposed solution (e.g., planned actions) and the expected outcomes (e.g., predicted minimum separation).

Similar to ATM, although UTM has not been fully established yet, some research has already started to explore how to increase the transparency of UTM based on their envisioned operational concepts and designed simulators [14, 59]. As UTM is expected to rely on high levels of automation, the transparency issue may be more urgent than it is for ATM. Without transparency, operators may struggle to understand the behaviour and limitations of automation, leading to a loss of situation awareness [14, 70]. Transparency research on one-to-many drone operations and multi-unmanned vehicle mission planning indicated positive effects of transparency, such as increased understanding and greater performance [24, 25, 178, 191, 192].

4.2.3. PROPOSED TRANSPARENCY TAXONOMY

To integrate the three perspectives on transparency in a pragmatic way in a path-planning context, a unified taxonomy is devised based on Figure 4.1, and is shown in Figure 4.2. Referring to the European Union Aviation Safety Agency (EASA) AI Roadmap [193], two fundamental concepts are proposed: *operational* transparency and *engineering* transparency. Operational transparency reveals (real-time) information about system status, goals, activities, and environmental impacts, directly supporting operational users in maintaining situation awareness, making informed decisions, and responding effectively

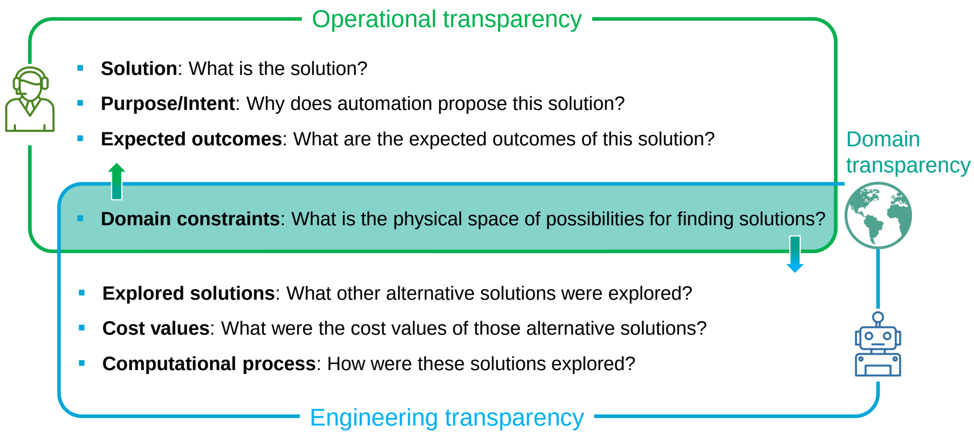


Figure 4.2: Proposed unified taxonomy for algorithmic transparency, integrating user-, model-, and ecology-centred perspectives.

to evolving conditions. Engineering transparency, in contrast, discloses system internal mechanisms, such as reasoning, exploration, evaluation, and decision-making processes, enabling users to develop a deeper understanding of system behaviour.

When operational transparency is fully revealed, operational users may have less need for engineering transparency [55]. This is because operational transparency equips users with the information they need to monitor and manage tasks effectively, reducing the necessity to understand the underlying engineering details. However, when unexpected events occur, such as automation failures, engineering transparency may become essential to help users identify the causes and alleviate the stress associated with confusion. Domain transparency, as a shared foundation in the middle, serves to connect the operational and engineering transparency. By clarifying the boundaries of feasible solutions, domain transparency helps users understand why the proposed solution (operational) and the exploration process (engineering) are both constrained within a certain range.

The transparency taxonomy contains seven categories, ranging from functional purpose and operational impact to operational boundaries and inner physical structure. This type of organisation is inspired by Rasmussen's Abstraction Hierarchy (AH) used in CWA and EID [53, 57, 153, 154]. The AH is structured based on typical human top-down, problem-solving strategies, starting at the desired system output (= functional purpose) and progressively descending towards the physical components of a system. As suggested by Springer and Whittaker [30], progressive disclosure may be needed for algorithmic transparency. In the proposed taxonomy, the transparency categories are organised hierarchically. From the Solution category to the Computational Process category, deeper algorithmic information is progressively revealed.

The Solution category emphasises that the algorithm's solutions should be clearly presented to users. For example, a path plan may include a sequence of actions, and users should be informed of when and where each action will take place. The Purpose/Intent category aims to reveal the algorithm's objective. This type of information will be complex if the algorithm has multiple objectives with varying weights. The Expected Outcomes category helps users assess the quality of the solution and decide whether to accept it or not. This decision depends not only on algorithm optimisations but also on user preferences. The Domain Constraints category lies at the intersection, forming solution spaces to explain the feasibility and robustness of solutions (operational) and also serving as a basis for system computation (engineering). The Explored Solutions category indicates the algorithm's exploration results in addition to the final optimal solution. The Cost Values category reflects the algorithm's criteria for evaluation and comparison. The Computational Process category represents the algorithm's underlying process for finding a solution. In search-based path planning, it mainly refers to the search process. Note that a transparency category is not entirely equivalent to a transparency level. A level could contain elements from one or more categories.

The proposed operational transparency categories can be regarded as a variant of the SAT model. As indicated by Bhaskara *et al.* [25], the projected outcomes (Level 3) in the SAT model may not necessarily represent a higher level, but rather a type of information, which is in line with the concept of the taxonomy. Different from the SAT model [22] and Lyons' human-robot transparency model [21], the proposed taxonomy is organised around *solutions*, with each category shedding light on a distinct aspect. This approach strengthens the interconnections between the various transparency categories, highlighting its hierarchical structure. The operational transparency categories also exhibit a correspondence with the engineering transparency categories, such as Solution to Explored Solutions, Purpose/Intent to Cost Function/Values, Expected Outcomes to Computational Process.

4.3. TRANSPARENCY DESIGN

As UTM encompasses a wide range of services [8, 9], this research mainly centres on tactical UTM operations in CTR around airports, in particular Rotterdam The Hague Airport. A web-based simulator DroneCTR[1] has been developed and improved as a test bed [14, 63]. The envisioned operational concept is similar to Dynamic Airspace Reconfiguration (DAR) [15, 16]. It assumes that CTR is assigned to UTM for navigating drones by default and operators can use geofences to block portions of the UTM airspace as required. As previous studies suggested a dedicated role for UTM supervision [15, 63], it also assumes that operators can only control drones rather than crewed aircraft. A centralised time-optimal path-planning algorithm is responsible for drone (re)routing to prevent entry into geofences and avoid conflicts with crewed aircraft. Therefore, this research is to reveal information about the UTM conflict-free routing service and the inner workings of the path-planning algorithm.

4.3.1. TRANSPARENCY ELEMENTS

Following the transparency taxonomy in Figure 4.2, a total of 20 transparency elements have been proposed for assisting the supervision of the automated UTM conflict-free routing service, as shown in Table 4.1. Like the proposed taxonomy, the transparency elements are primarily based on established design practices from previous studies, high-

[1] URL: `http://dronectr.tudelft.nl/`, ID: demo

Table 4.1: Proposed transparency elements.

Transparency Category	**Transparency Element**
Solution	The proposed (new) path and old path Estimated state and planned action at each waypoint
Purpose/Intent	The underlying goals and intentions of the system (e.g., minimising flown track miles)
Expected outcomes	If the drone follows the old path • Predicted location of separation loss • Predicted start time of separation loss • Predicted minimum separation • Predicted probability of separation loss If the drone follows the proposed (new) path • Predicted location of CPA • Predicted time to CPA • Predicted minimum separation • Predicted probability of separation loss
Domain constraints	Safe separation standards between aircraft **Manoeuvring space**: the flight range governed by battery power and environmental conditions **Flight mission boundary**: certain drones can only fly within a pre-approved area **Wind field**: wind speed and direction
Explored solutions	**Search graph**: a search graph is how automation discretises a continuous space, and the generated path can only follow the edges of the graph **Explored nodes**: explored potential waypoints **Search trees**: explored potential paths
Cost function/values	The cost values of the explored potential paths given the system's goals and intentions
Computational process	**Search process**: a dynamic process that indicates how to generate the path

lighting that this work complements rather than replaces them.

In terms of operational transparency, the Solution category contains two elements: the old path and the proposed (new) path [55, 194, 195]. The old path is the path the drone followed before rerouting, which hints at why the drone needed to reroute in the first place (e.g., due to a conflict). A path is essentially built from a sequence of states and actions. To gain a deeper understanding of the proposed path (solution), the estimated states and planned actions should be clearly revealed (e.g., where certain heading/altitude changes will take place). The Purpose/Intent category can be presented by text-based explanations [61, 196]. In this case, the path generated by the UTM system aims to be time-optimal and conflict-free. For the Expected Outcomes category, two different "what-if" situations are considered: what if the drone continues following the old path, and what if it flies along the proposed (new) path. To observe the outcomes of the paths, four metrics were proposed referring to other research [55, 60]: predicted location of separation loss (and predicted location of CPA), predicted start time of separation loss (and predicted time to CPA), predicted minimum separation and predicted probability of separation loss. Regarding the Domain Constraints category, a range of restrictions linked to drone endurance and no-fly zones, such as drone manoeuvring space and flight mission boundary, were presented [14, 107]. The wind field was incorporated as well since drones are susceptible to wind [197].

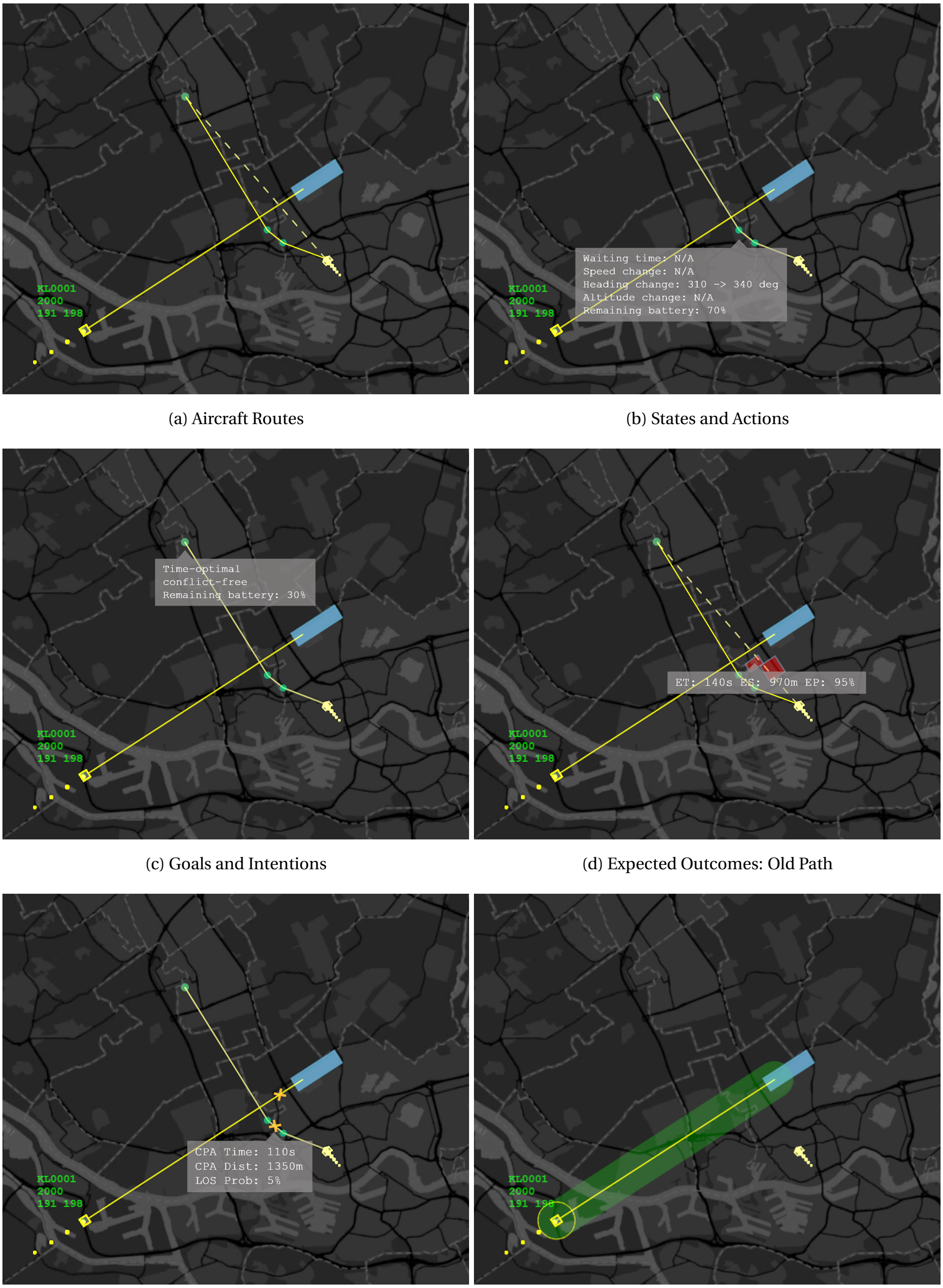
In terms of engineering transparency, the domain constraints, such as the manoeuvring space, also limit the search space of path planning, illustrating why the system only searches within a certain area [198]. For the Explored Solutions category, three elements were proposed with reference to existing pathfinding visualisers [43, 44, 48, 100]: search graphs, explored nodes and search trees. These three elements could also be simultaneously showcased to convey information that is more meaningful and integrated. The cost function/value is somewhat similar to the expected outcomes, with both utilising specific metrics for computation. However, the cost function represents the goals of the system, while the outcomes are future projections of the solution. The cost function in this study optimises only a subset of factors, such as flight efficiency (time-optimal), without considering environmental uncertainty (e.g., optimising for robustness). The Computational Process category reveals the algorithm's dynamic search process (can be achieved through animation [41, 42, 155]), providing more details about the algorithm's expansion of search nodes and search trees.

4.3.2. VISUAL PROTOTYPES

The corresponding visual prototypes for the proposed transparency elements have also been developed, as shown in Figures 4.3-4.5. The proposed (new) path is a solid yellow line that the drone will follow, while the old path is drawn as a dashed yellow line that can be hidden if desired. The waypoints of the proposed path are presented as green dots and more details regarding the waypoints can be accessed, such as waiting (loitering) time, speed/heading/altitude change and remaining battery. The destination of the drone is also represented as a dot, containing information about the goals and intentions of the drone (path-planning algorithm).

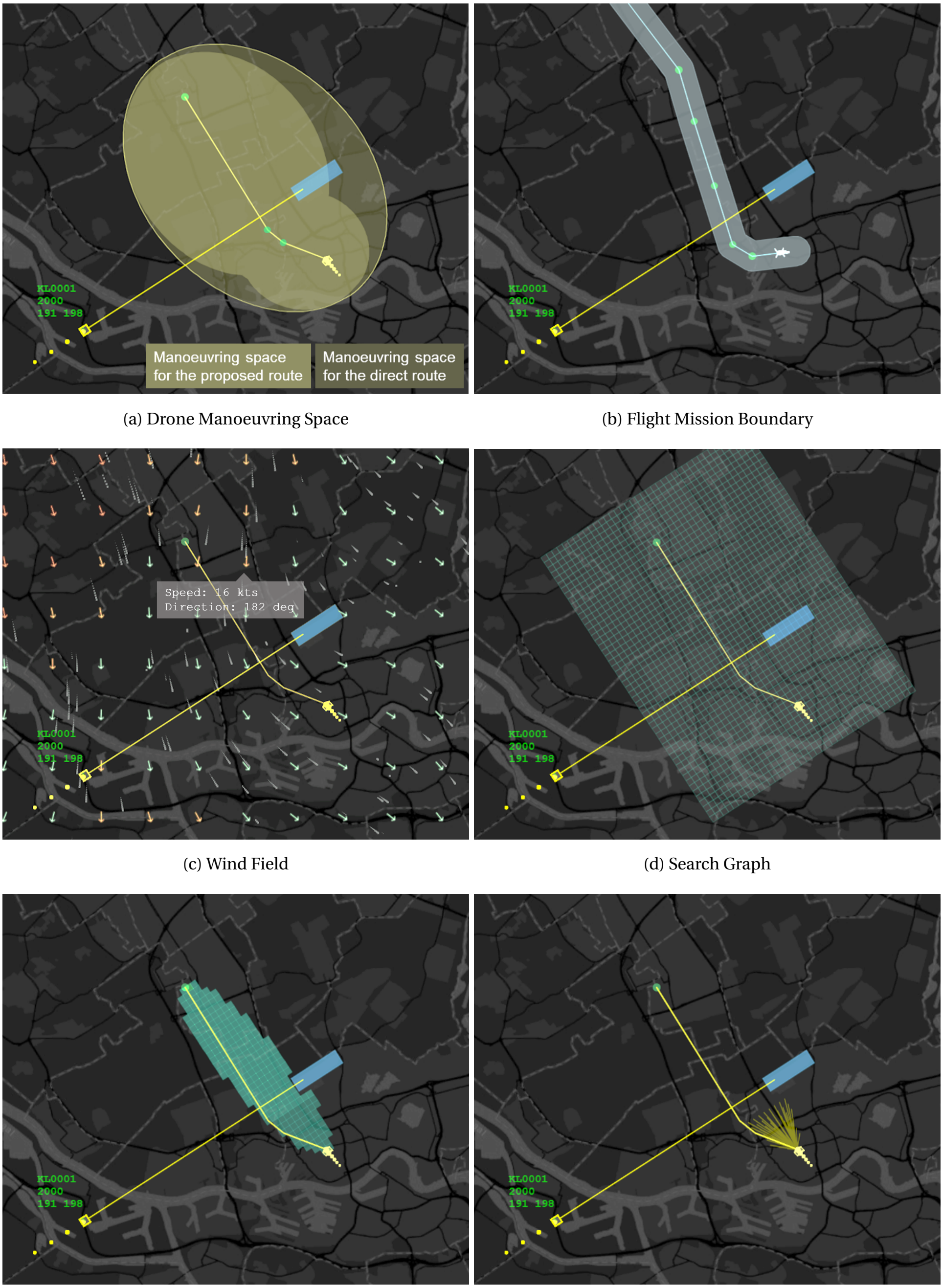
For the Expected Outcome category, the proposed metrics can be collectively showcased, allowing the depiction of the expected outcomes with only two images. One is for the conflict situation (old path): the red area indicates the predicted location of separation loss. Detailed information about the conflict can be displayed, including estimated start time of separation loss (ET), estimated minimum separation distance (ES) and estimated probability of separation loss (EP). The other one is for the normal situation (new path): the amber crosses indicate the predicted locations of CPA. The predicted time to CPA (CPA Time), predicted distance at CPA (CPA Dist) and predicted probability of separation loss (LOS Prob) can also be accessed via the crosses.

For the domain constraints, the safe separation is denoted by a yellow circle around crewed aircraft. It means that if a drone flies into this circle, there will be a loss of separation. Combined with the route, a green air corridor for crewed aircraft can be presented. The drone manoeuvring space is a visual representation of the drone's range governed



(a) Aircraft Routes
(b) States and Actions
(c) Goals and Intentions
(d) Expected Outcomes: Old Path
(e) Expected Outcomes: New Path
(f) Safe Separation Standards

Figure 4.3: Prototypes showcasing the proposed transparency elements (Part 1).



(a) Drone Manoeuvring Space

(b) Flight Mission Boundary

(c) Wind Field

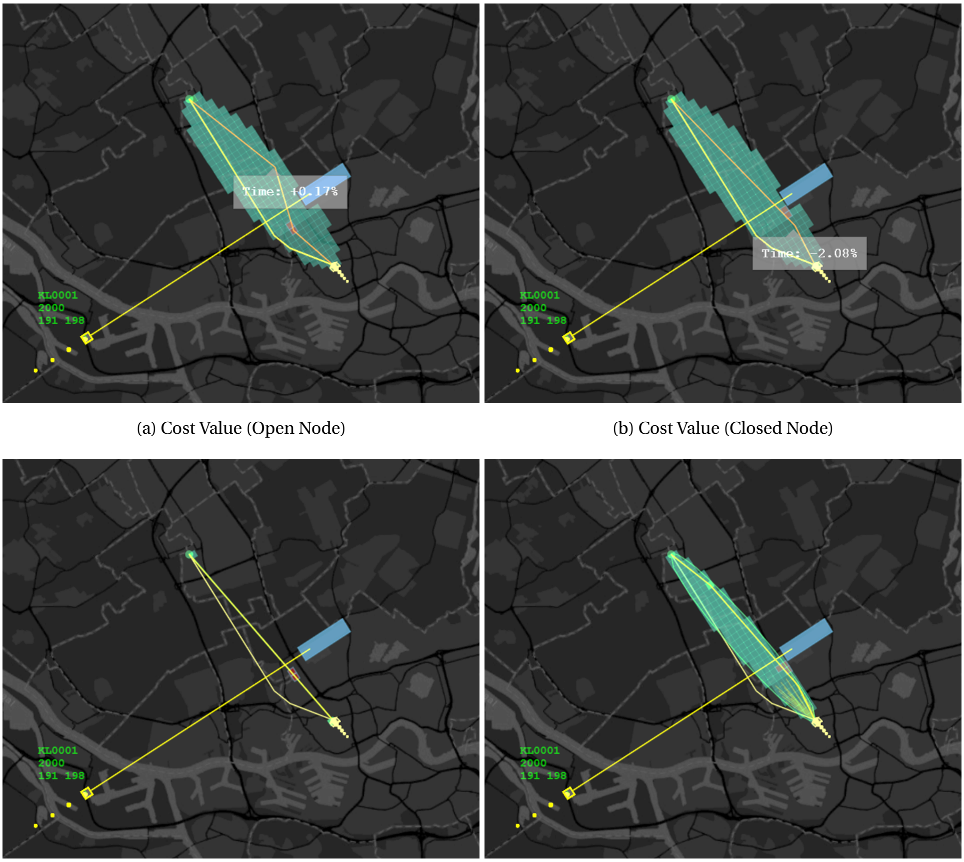
(d) Search Graph

(e) Explored Nodes

(f) Search Tree

Figure 4.4: Prototypes showcasing the proposed transparency elements (Part 2).

4



(a) Cost Value (Open Node) (b) Cost Value (Closed Node)

(c) Search Process (Animation 1) (d) Search Process (Animation N)

Figure 4.5: Prototypes showcasing the proposed transparency elements (Part 3).

by battery power and environmental conditions such as wind. Generally, a narrow manoeuvring space corresponds to low excess battery power and/or increased headwind conditions. Two different types of manoeuvring space are presented in the figure: the dark yellow indicates the proposed route and the bright yellow denotes the direct route. The mission boundary is designed as an air corridor in this case and the drone can only fly within this area. Wind field information can not only be shown by speed and directional values, but also in the form of animated particles and/or coloured arrows. Different colours indicate different wind speeds: green denotes winds that have relatively little impact on drones while red represents strong winds that could cause drones to drift.

A grid-based time-optimal path-planning algorithm [85] is employed for the UTM conflict-free routing service. Therefore, the search graph in this case is a grid bounded by the search space, which is determined by the drone manoeuvring space and, if applicable, the mission boundary. The explored nodes are the green cells that are potential waypoints explored by the algorithm while the search tree represents all potential paths that have been explored. The "heuristic" (direct) lines between the branch ends and the destination node have been hidden to avoid visual clutter. Note that the potential paths explored by the algorithm are just promising to be time-optimal and conflict-free (the algorithm's purpose). During the search process, the algorithm produces (many) intermediate results that failed to be the final solution, because they are not optimal and/or are unsafe. The cost value of a node, representing the cost of an explored path passing through it, can be retrieved. The search process is a dynamic process that indicates how the search tree is composed and how the final path was found. Similar to [43, 44, 48], the dynamic search process can be demonstrated through (fast-time) animation.

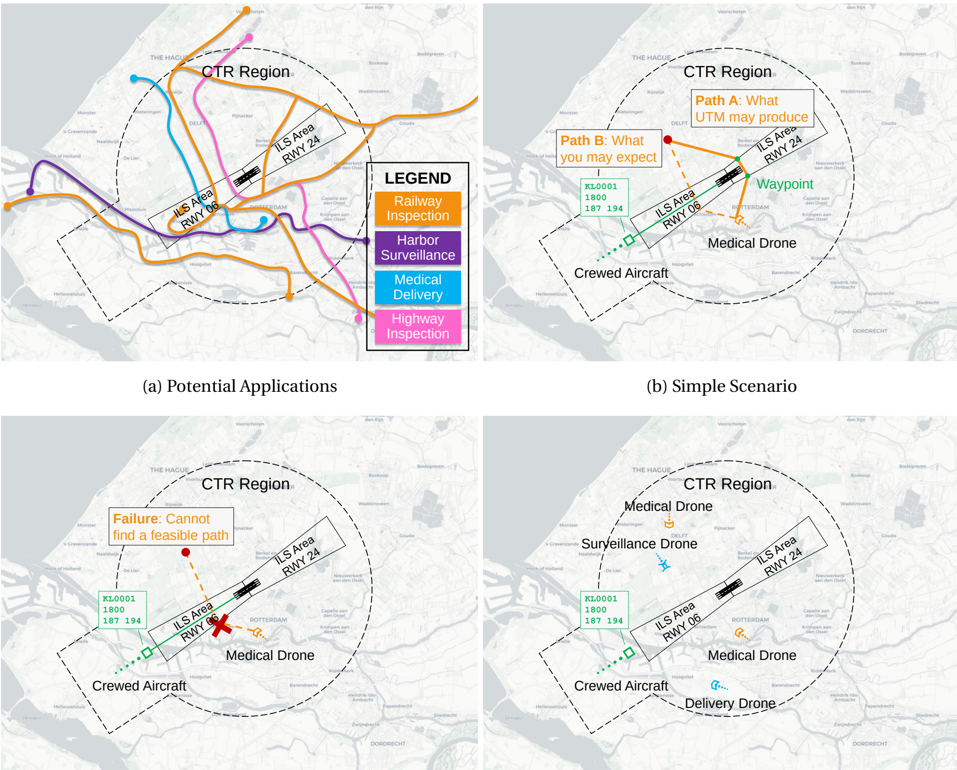
4.4. Methodology

4.4.1. Overview

Based on the proposed transparency elements and visual prototypes, a questionnaire was designed to investigate the transparency needs of operators for supervising the UTM system. The overall goal of this survey study is to validate the content and structure of the proposed transparency taxonomy in the tactical UTM context. If all the devised elements are deemed valuable for UTM supervision, then every category in the taxonomy is indispensable. The survey study also examines how operators categorise these transparency elements, exploring whether three distinct types of transparency emerge from operators' viewpoints.

At the start of the questionnaire, participants were given a detailed explanation of the background and a hypothetical operational concept, recognising that they may have different visions of future UTM operations. Some personal information was then collected to aid data analysis. This study was approved by the Human Research Ethics Committee (HREC) under number 3374.

Given the primary focus of this study on UTM within CTR, Rotterdam-The Hague Airport was selected as a use case. The potential drone applications in the airport's vicinity, such as railway and highway inspection and medical delivery, are illustrated in Figure 4.6a. Three distinct hypothetical scenarios, as depicted in Figure 4.6b-4.6d, were presented to stimulate participants' thoughts: a simple scenario encompassing only a single drone, a failure scenario entailing an automation failure case and a complex scenario involving multiple drones with diverse missions and types. For the simple and failure scenarios, a trajectory-contrastive question and a failure question [76, 115] were provided for further inspiration: 1) why path A rather than path B, and 2) why the system fails. For the complex scenario, time pressure issues may be more salient [60] and the usefulness of transparency information for supervision might be different.



(a) Potential Applications (b) Simple Scenario

(c) Failure Scenario (d) Complex Scenario

Figure 4.6: Schematic diagrams for operational scenarios.

To enhance the authenticity of the scenarios, descriptive text for the three operational scenarios was also provided before the rating began. For the simple and failure scenarios, "*a medical drone needs to pass through an area near the runway to deliver emergency supplies between two hospitals (from Rotterdam to Delft) as quickly and as safely as possible, but a crewed aircraft is about to land. The automated UTM conflict detection service has detected a potential separation problem (= conflict) between these two aircraft*". For the complex scenario, "*when more drones need to cross the area covering the extended runway centreline, the conflict scenario may become more complex. Such increased complexity may have an impact on your transparency needs in the light of 'information overload'. Note that the UTM system in this investigation only deals with separation conflicts between crewed and uncrewed aircraft. The collision avoidance among drones is assumed to* *be achieved by 'sense and avoid' systems onboard drones. Therefore, the automated conflict detection & resolution services for drones only regard crewed aircraft as dynamic obstacles".*

4.4.2. QUESTIONNAIRE STRUCTURE

There were two main phases in the questionnaire, as presented in Figure 4.7. In the first phase, the proposed 20 transparency elements derived from the unified taxonomy were presented as response options, with their order being randomised. Participants were asked to rate the elements using a 5-point Likert scale according to their perceived usefulness for understanding and supervising the automated UTM conflict-free routing service. Open-ended questions were also present to inquire the reasoning behind their ratings. To obtain the original opinions of participants and avoid any potential bias in their results, the visual prototypes were deliberately withheld from participants in the first phase.

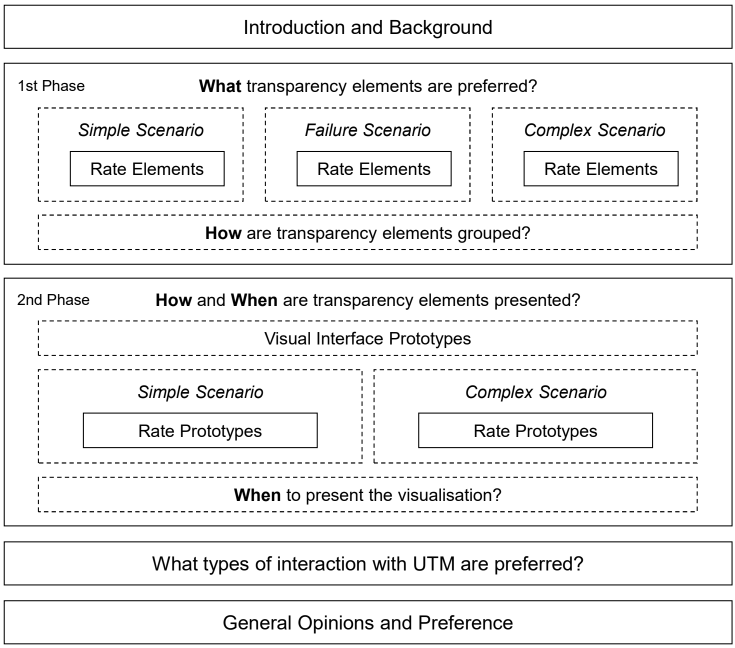


Figure 4.7: The structure of the questionnaire

Then, the questionnaire explores how participants proposed to group transparency elements that belonged together in their opinions. This can offer valuable insights into what transparency categories or elements should be connected and/or presented together in practice. The groups identified by operators could also help validate whether the proposed unified taxonomy is reasonable from the operators' perspective, specifically whether there are indeed three distinct types of transparency. Considering that each participant may have their own groups of transparency elements, a weighted adjacency matrix is employed to summarise their preferences. The weight here refers to the number of times two elements are divided into the same group. Then, based on this adjacency matrix, a weighted graph can be constructed to visually depict the interconnections among various transparency elements. Finally, to group these elements (i.e., the vertices of the weighted graph), the Walktrap community detection algorithm [199] will be applied, as illustrated in Figure 4.8.

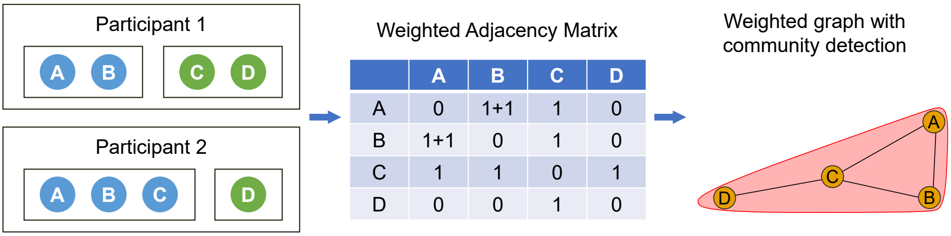


Figure 4.8: Data processing of the grouped transparency elements

In the second phase, the visual prototypes for the proposed transparency elements were presented and then 5-point Likert scale questions were provided to inquire the usefulness of the prototypes for understanding and supervision in the simple and complex scenarios again. To limit the questionnaire length, participants were provided only with examples that successfully generate paths, as shown in Figures 4.3-4.5, and thus the failure scenario was omitted in this phase. Previous studies indicated that user preferences on transparency may change after actual experience with it [30, 60]. This phase can assess whether participants altered their opinions after viewing the prototypes.

When to present each prototype was also investigated after then. For example, operators could enable the UTM system to automatically determine the timing of information presentation (adaptive). Alternatively, operators could manually show or hide the elements by clicking on relevant buttons (adaptable). After rating the transparency elements and prototypes, participants were asked to select and rank their preferred types of interactions with the UTM system. Finally, the questionnaire ended with some general questions on transparent UTM systems in terms of importance, additional workload and acceptance concerns.

4.4.3. PARTICIPANTS

Previous research indicated that UTM supervision may not be appropriate for ATCos to perform besides their regular ATC tasks and a dedicated UTM supervisor may be necessary [15, 63]. Since the role of UTM supervisors differs from ATC such as tower control, they are not required to undergo the same training and licensing as ATCos. Instead, they could be someone more familiar with drone operations, like drone pilots or drone engineers. Therefore, both ATCos and drone experts were invited to participate in this study. The results could offer a more comprehensive insight into the transparency needs of operational users. Additionally, given the diverse professional backgrounds of ATCos and drone experts, their transparency needs may also be different [33, 61, 111].

Twenty four operators from Europe and China volunteered to participate in this survey of which twelve were licensed ATCos (e.g., Rotterdam and Shanghai controllers) and twelve were drone experts (e.g., drone engineers from TU Delft and drone pilots from companies). Their experience in ATC and drone operations is summarised in Figure 4.9. One participant who serves as both an ATCo and a drone expert was classified as an ATCo in this survey.

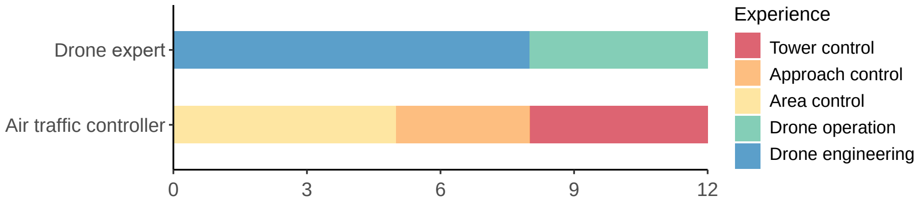


Figure 4.9: Participants' experience in air traffic control and drone operations.

4.5. RESULTS

4.5.1. GENERAL OPINIONS ON TRANSPARENCY

Figure 4.10 presents the general opinions of participants on transparency after filling out the questionnaire. Most participants believed that transparency plays an important role in supervising the UTM system and will significantly influence their level of acceptance and trust. One ATCo held the view that transparency would not affect acceptance at all, because his/her main concern was about the number of aircraft in flight. One drone expert believed that transparency would have a slight impact on trust, because he/she would trust the UTM system overall once it is fully operational. The additional workload that transparency could bring is considered to be relatively manageable. Over half of the participants thought that the additional workload would not be very high. This result should be interpreted with care, because this study did not feature a real-time, interactive human-in-the-loop simulation with dynamic traffic situations. As such, conclusions about transparency-induced workload warrants further research.

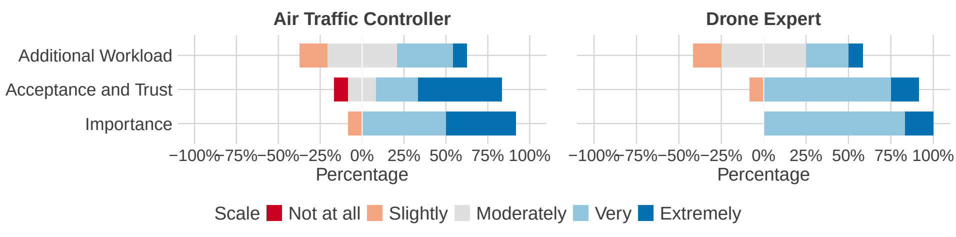


Figure 4.10: General opinions on transparency.

4.5.2. PREFERRED TRANSPARENCY INFORMATION

DATA ANALYSIS AND STATISTICS

By converting the Likert scale ratings into numerical values, statistical analysis could be performed to assess the impact of different factors on operators' transparency needs.

Wilcoxon Signed-Rank tests were conducted for an overall comparison of operational and engineering transparency. The matched-pairs rank biserial correlation coefficient rc [200] was then calculated to measure the effect size for Wilcoxon Signed-Rank tests (small ≥ 0.1, medium ≥ 0.3, large ≥ 0.5). Friedman tests were performed to further analyse the differences in operational and engineering transparency among various scenarios, followed by Exact tests [163] with the Bonferroni correction for pairwise comparisons. Kendall's coefficient of concordance w was used to measure the effect size for Friedman tests (small ≥ 0.1, medium ≥ 0.3, large ≥ 0.5). To compare the differences between ATCos and drone experts, Mann-Whitney U tests were conducted. The effect size r was calculated for Mann-Whitney U tests (small ≥ 0.1, medium ≥ 0.3, large ≥ 0.5), which is defined by the standardised test statistic z from the tests divided by the square root of the total number of observations. The significance level was set to 0.05. As the effect size reflects the magnitude of the difference between groups [164], it is ideal to have both a statistically significant result ($p < 0.05$) and a large effect size to claim a clear and meaningful difference. A low effect size suggests that the statistical significance should be interpreted with caution.

COMPARISON OF OPERATIONAL AND ENGINEERING TRANSPARENCY

The Likert scale ratings for the transparency elements and visual prototypes are shown in Figures 4.11 and 4.12. The failure scenario was not included in the second rating phase with the visual prototypes and the reason has already been mentioned in the previous section. Generally, all proposed elements were considered valuable for supervising the UTM system, although some of them may have limited utility in some scenarios.

Compared to operational transparency, engineering transparency is considered less useful in the tactical UTM operations, as expressed by an ATCo: "*I need it to tell me why it gives this route and the disadvantage of this route. I don't think how it finds this route is useful*". Drone experts had similar views: "*I would be most interested in knowing when, where and how the conflict might occur from the system's point of view … I need to access objective metrics from which I can verify the soundness of the proposals. I do not want to be bothered by the inner workings of the system (e.g. how the search is conducted) since I feel it may lead to an information overload.*" These arguments are consistent with the SESAR projects reviewed in this chapter, which focuses on the goals and intentions of systems and the expected outcomes of solutions. Additionally, a drone expert remarked: "*It has to be simple during actual operations … the operational environment might be over-engineered - these items should be more of things to revisit in hindsight*". Interestingly, operational and engineering transparency were not mentioned in the questionnaire, but judging from the results, participants seemed to distinguish between them very well.

The operational transparency encompasses two distinct categories of expected outcomes: one pertaining to the proposed (new) path and the other to the old path. A drone expert suggested that "*a really simple table was needed to compare the main elements of two paths*". This comment shares similarities with the TAPAS project which also utilises tables to present various metrics. The expected outcomes of the old path are also considered relatively less useful. One expressed it as follows: "*I think the old path is not necessary for avoidance. The current states of both manned and unmanned aircraft and their predicted paths are more important*". Another drone expert also remarked: "*The predicted states based on the proposed path matters more than the old path*". This is probably

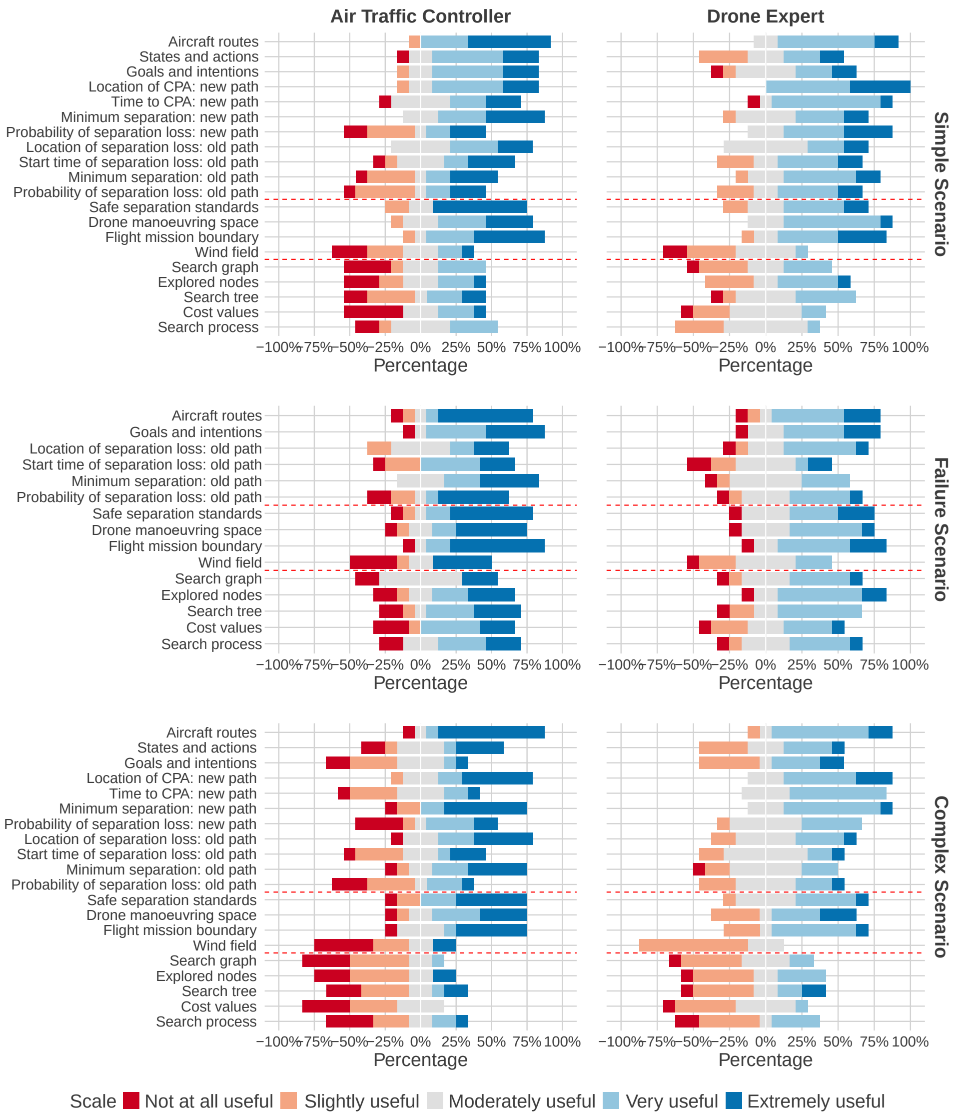


Figure 4.11: Likert scale ratings for the proposed transparency elements. The red dashed lines denote the operational, domain and engineering transparency categories. The transparency elements from the "Solution" category are absent in the failure scenario because there is no solution in this case.

because the proposed path is more relevant to the current situation.

As for the domain constraints, although a drone expert pointed out that "*Large wind or stormy weather will create critical situations for aircraft, especially drones*", the wind field is generally considered least useless compared to other constraints. A possible reason is that the wind field only presents basic environmental information, which is not di-

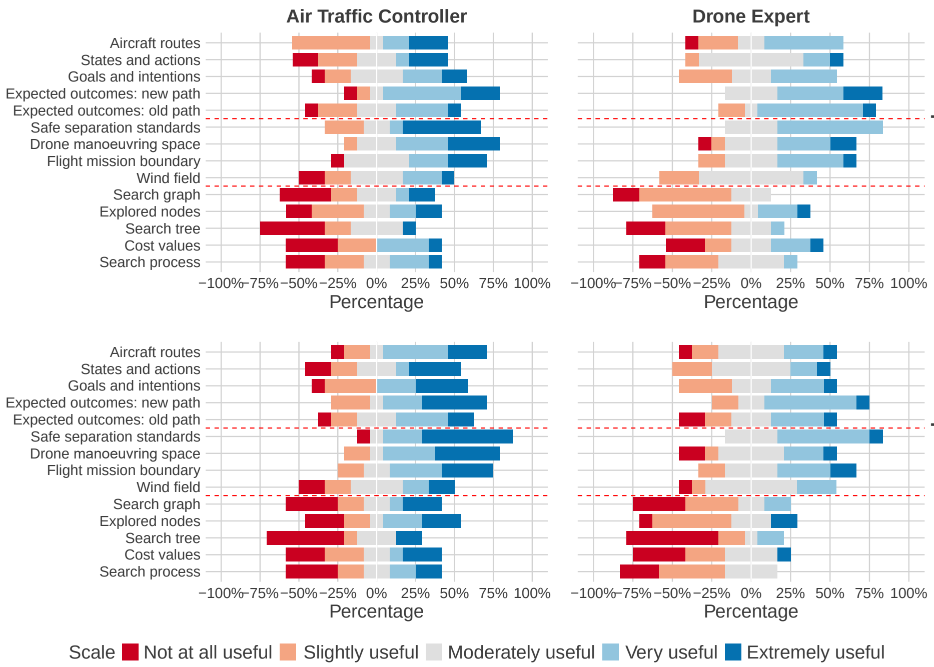


Figure 4.12: Likert scale ratings for the visual prototypes.

rectly associated with the goals of operators. It may be more effective to introduce no-fly zones determined by wind conditions, taking into account both wind speed and drone performance. In other words, presenting wind information in terms of how it impacts drones is considered more useful than simply presenting the wind condition itself.

For a clearer comparison, average ratings are computed for operational and engineering transparency in different scenarios, as depicted in Figure 4.13. Based on the average ratings of the two phases, Wilcoxon Signed-Rank tests revealed significant differences between operational and engineering transparency in both ATCo ($V = 74$, $p < 0.01$, $rc = 0.897$) and drone expert ($V = 63$, $p < 0.01$, $rc = 0.909$) groups.

PREFERENCES IN DIFFERENT SCENARIOS

Figures 4.11-4.13 also indicate that differences exist not only between the types of transparency but also among the scenarios. In the failure scenario, engineering transparency is deemed more useful compared to other scenarios. This is probably because operators need more information about the system's internal process to figure out what happened inside the system. The information concerning constraints could be particularly helpful: "*If there's no good solution, this should come from some limitations from the dynamics of drones*". "*The waypoints, manoeuvring space, and boundaries are the key to finding the desired path*". Actually, some participants indicated, "*that everything allowing to understand why the system fails is useful*". However, it is worth noting that the occurrence of failure scenarios should be minimised as much as possible. *Robustness* was repeatedly

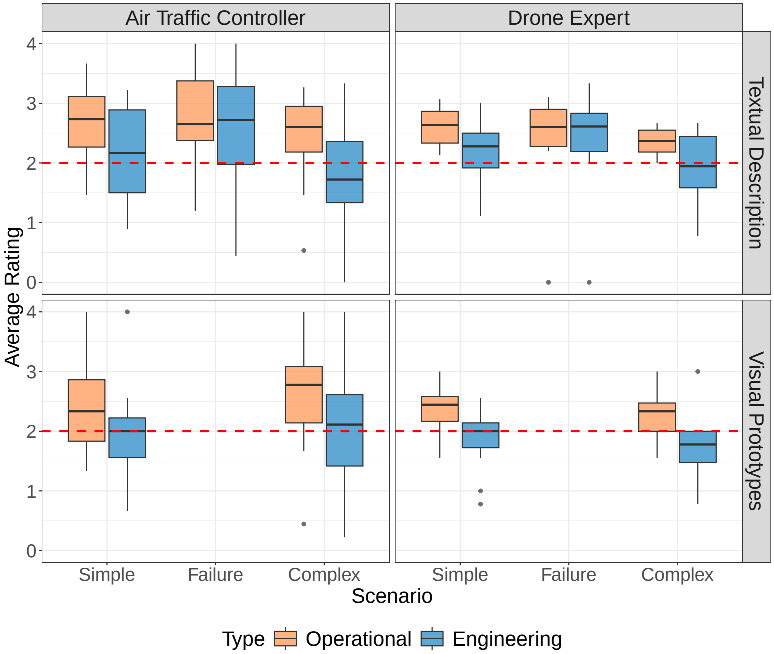


Figure 4.13: Average ratings for operational and engineering transparency based on "Not at all useful = 0" to "Extremely useful = 4". Domain transparency is included in both operational and engineering transparency

mentioned as one of the crucial factors affecting their acceptance of a highly automated UTM system. A participant stated, "*If there is no feasible path, it should never cross a route with manned traffic*". In the complex scenario, engineering transparency is considered even less useful, as "*too much information could overwhelm operators.*" A respondent suggested that "*it is more important to only look at the conclusive information*". The information indirectly related to safety and situations should probably be hidden in the first place.

To further confirm these differences, a statistical analysis was performed based on Figure 4.13. For the ratings with only textual description, Friedman tests revealed significant differences among conditions (Three Scenarios × Two Transparency Types) in the ATCo group ($\chi^2(5) = 23.002$, $p < 0.01$, $w = 0.383$), but no such differences were observed in the drone expert group. For the ATCo group, pairwise comparisons with the Bonferroni correction further revealed that the "Complex-Engineering" condition was significantly different from the "Simple-Operational" ($D = 33.5$, $p < 0.01$), "Failure-Operational" ($D = 35.0$, $p < 0.01$) and "Failure-Engineering" ($D = 30.5$, $p < 0.01$) conditions. For the ratings with visual prototypes, Friedman tests revealed significant differences among conditions in both ATCo ($\chi^2(3) = 9.083$, $p = 0.028$, $w = 0.252$) and drone expert ($\chi^2(3) = 19.817$, $p < 0.01$, $w = 0.550$) groups. However, pairwise comparisons did not confirm significant differences in the ATCo group between conditions. It can also be observed in Figure 4.13 that the data spread in the ATCo group is large. For the drone expert group, pairwise comparisons with the Bonferroni correction further found that the "Complex-Engineering" condition was significantly different from the "Simple-Operational" ($D = 24$, $p < 0.01$) and "Complex-Operational" ($D = 18$, $p = 0.027$) conditions.

PREFERENCES WITH VISUAL PROTOTYPES

To further explore whether participants' preferences changed after viewing the visual prototypes, Figure 4.14 shows the relationships between the average ratings for the two rating phases. Overall, participants' preferences remained relatively consistent, suggesting that the visual prototypes in the second phase aligned with their expectations formed through the textual descriptions in the first phase. However, there are still some notable changes. After viewing the visual prototypes, ATCos found the transparency information less beneficial in simple scenarios than previously thought (negative change), but more beneficial in complex scenarios (positive change). In contrast, drone experts regarded the transparency information as less useful in both simple and complex scenarios than initially expected (negative changes). The negative changes for participants are likely driven by concerns about visual clutter and a preference for maintaining a clean interface. The positive change for ATCos may stem from their experience with ATC interfaces. They may believe that more supportive information is required in complex scenarios and that the visual prototypes can be integrated into a single interface for support without excessive visual overlap.

COMPARISON OF ATCOS AND DRONE EXPERTS

Generally, the needs for different types of transparency were found to be similar between ATCos and drone experts. Mann-Whitney U tests did not reveal any significant difference between the two operator groups. However, as evident from Figures 4.11 and 4.12, some minor distinctions still exist on specific elements. Among the four metrics indicating the expected outcomes, the probability of separation loss is found to be favourable by drone experts: "*I may pay more attention to … the predicted probability of separation loss.*" The probability would indicate the uncertainties of the system. If the system's confidence in resolving conflicts is not high enough, operators may need to intervene in the system. However, as an ATCo stated, "*ATC does not control considering probability*". Also, another ATCo expressed: "*To some extent, probability may not represent its level of danger very well. If I realised the separation was not enough, I thought my priority was to increase the separation to prevent it, not just to compare the probability*". In fact, the automated UTM conflict-free routing service should be robust enough to reduce the probability of separation loss to "zero" in most cases. When the probability is not zero, the system should provide some additional explanations to indicate its limitations. For example, changes in wind conditions could lead to variations in flight duration, thereby increasing the probability of separation loss and triggering new conflicts. Furthermore, ATCos also emphasise the information regarding predicted locations of separation loss and CPA and predicted minimum separation. As mentioned by an ATCo, "*Two elements are of utmost importance: which location will the separation loss be and to which location does it shift when a new route is proposed.*" This preference can be clearly observed in the complex scenario (see Figure 4.11).

4

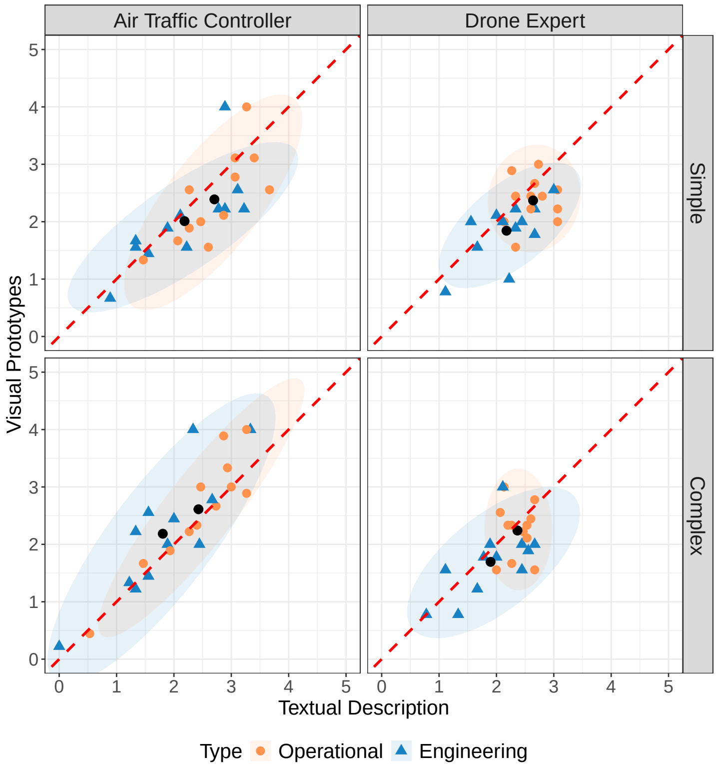


Figure 4.14: The relationships between the average ratings for the two phases: one with only textual descriptions and the other with visual prototypes. Confidence ellipses (95%) are presented per transparency type, with black dots indicating their centres.

As shown in Figure 4.13-4.14, there is a notable discrepancy in the variance of ratings between ATCos and drone experts. It appears that ATCos tend to be more forthright and confident, often expressing their views at either end of Likert scales. There also seems to be a disagreement among ATCos, resulting in the increased variance. This phenomenon primarily exists within the tower and area controller groups. Drone experts, in contrast, tend to hold more conservative views, leaning towards the neutral side. There appears to be more consensus among drone experts. Since the sample size is not large, more data would be needed to substantiate this observation.

Additionally, ATCos expressed a greater preference for engineering transparency elements than drone experts in complex scenarios after viewing the visual prototypes (see Figures 4.12 and 4.13). One possible reason is that ATCos generally take a more critical view of automation [201] and thus may seek as much information as possible to audit it. In comparison, drone experts, being more familiar with automation, may be more biased to accept and trust UTM (as noted by one drone expert shown in Section 4.5.1), and thus may require less engineering transparency information.

4.5.3. TRANSPARENCY ELEMENT GROUPING

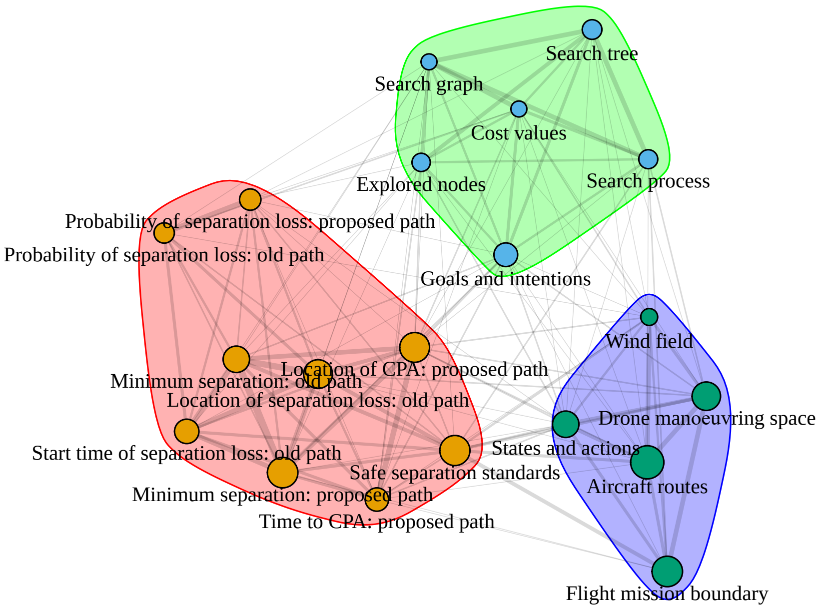
Based on the weighted adjacency matrix and the Walktrap community detection algorithm, the correlations between the proposed transparency elements are computed, as shown in Figure 4.15. Both ATCos and drone experts categorise these elements into three groups, with their results being nearly identical. The sole distinction lies in how the safe separation standard is allocated: for ATCos, it is associated with the expected outcomes (red group) whereas for drone experts, it is linked to domain constraints and solutions (purple group). This is possibly because the goal of ATCos is to ensure that the outcomes meet the established separation standards. The safe separation can be regarded as a baseline or minimum requirement, which is often presented in ATC decision-support tools.

In summary, the groups classified by operators can be labelled as follows: Expected Outcomes (red), Solution & Solution Space (purple) and Internal Process (green). This can be regarded as a more condensed variant of the proposed taxonomy. In the green group, the goals and intentions are closely connected to the cost values since cost functions should typically be designed in accordance with goals. The correlations among the proposed transparency elements can provide guidance and reference for further devising transparency levels and models, as they illustrate which elements operators prefer to see concurrently for understanding and supervision.

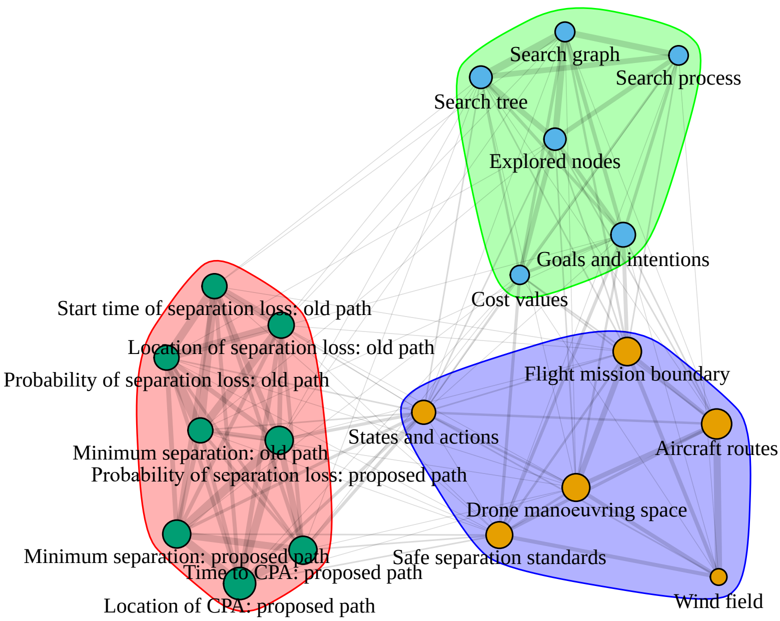
4.5.4. INTERACTION AND INTERVENTION

Figure 4.16 illustrates the trigger conditions preferred by operators for the transparency elements. Overall, there is no universally agreed-upon trigger condition for each transparency element. The operators have their individual preferences for determining when to present the transparency information. It seems that ATCos generally prefer to click on (or hover over) aircraft, while drone experts tend to favour global or automatic activation on demand. This is probably due to their different experiences with ATC and drone operations. For ATCos, they are more accustomed to interacting with radar screens where information and actions are typically associated with each aircraft, including radar labels, speed vector lines and history dots. In contrast, for drone experts, they may rely more on automation to assist in planning trajectories and avoiding obstacles. More than 25% of the ATCos opted to never present the inner workings of the algorithm, likely to maintain a clean interface, which is in line with their ratings on the transparency elements (See Figures 4.11 and 4.12). Conversely, drone experts prefer retaining the option to access more additional information.

With transparency information, operators may be able to understand the current system state and maintain situation awareness. However, they also need to know what actions they can take if something goes wrong; otherwise, human supervision of the system would be pointless. Sometimes, the generated path may not align with operators' expectations or preferences on how to resolve conflicts. The UTM system should incorporate interaction methods that allow operators to intervene when required or desired. Therefore, during the survey, in addition to rating the elements, participants were also asked to select and rank different interaction methods with UTM. The results are presented in Figure 4.17. Interestingly, according to the Rank 1, more than 50% of the ATCos and drone experts prefer active control over drones instead of passive control (geofence activation) or mixed control (waypoint constraints on algorithms). This observation is in line with



(a) Air Traffic Controller



(b) Drone Expert

Figure 4.15: The correlations between the proposed transparency elements. The vertex size corresponds to the average rating.

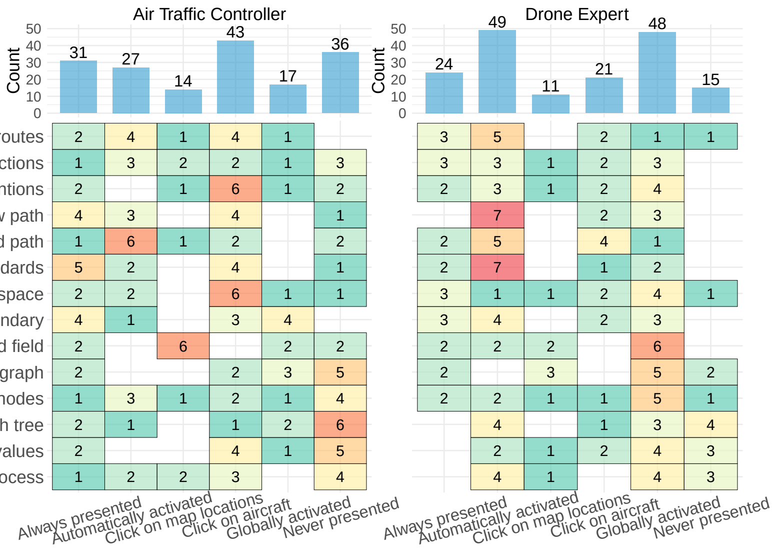


Figure 4.16: Trigger conditions of the visual prototypes.

previous human-in-the-loop experiments in dynamic UTM scenarios [14, 15, 63]. As the scenario becomes more complex, ATCos tend to prefer passive control to protect crewed aircraft from drones whereas drone experts favour active control to navigate drones manually. This is probably because of their different professional backgrounds and experiences with drone operations. Drone experts may be more confident in taking control of drones to address issues, whereas ATCos prioritise maintaining the safety of crewed aircraft by clearing their paths of any obstacles.

4.6. DISCUSSION

4.6.1. OPERATIONAL AND ENGINEERING TRANSPARENCY

The results of the user study revealed that operational transparency is more useful than engineering transparency for tactical UTM operations, as recognised by both ATCo and drone expert groups. This finding aligns with the previous transparency research in ATM [55, 60, 61]. For example, TAPAS [55] also introduced the concept of operational transparency in their research, defining it as the provision of operational information driving decisions with respect to operators' pragmatics constraints. They believed that providing information that maintains operators' situation awareness is sufficient to develop trust in AI, even in high-stakes fields like ATC.

This conclusion appears to contradict the prevailing perspective in XAI research that aims to open "black boxes" (engineering transparency) to increase human understanding, trust and acceptance of AI-based systems [31, 33]. However, actually, there is no contradiction between them. As indicated by Springer and Whittaker [30], Kizilcec [147], trust is affected by *expectation violation*. For ATM and UTM, the operators' expectations are to ensure safe and efficient operations. As long as the system's proposed solution

Figure 4.17: Ranks of interaction methods with the UTM system.

can meet this goal, operators will probably accept and trust it. In this case, operational transparency is to help operators evaluate the proposed solution within a specific context [202] and thereby maintain their situation awareness. There is almost no need for operators to access engineering transparency in normal scenarios, especially when operators have extensive operational experience like ATCos [55]. Operators' transparency needs may diminish over time as they become increasingly familiar with the system.

However, as this study suggests, in the case of automation failure, operators tend to want more engineering transparency to understand what happened deeper inside the system. This is precisely because of the expectation violation. The automated system did not work as expected, resulting in a reduced trust and an increased demand for explanations and engineering transparency [74]. Therefore, to effectively address all possible situations, both operational and engineering transparency is important. Certainly, it does not mean that all information is required to be presented simultaneously. To avoid overwhelming operators, transparency should be provided on demand [30].

4.6.2. Potential improvements to transparency design in UTM

Section 4.3 shows the design of twenty transparency elements and fourteen corresponding visual prototypes. Based on the study results, they can be further improved.

In operational transparency, some operators considered the old path relatively less important, possibly because drones may frequently adjust their paths in dynamic environments, causing the old path to change often as well. Focusing on the old path may provide limited value for monitoring the current situation and system state. Therefore, it may be better to present the drone's original plan as the "old path", allowing operators to understand how the new path deviates from the original plan. For a point-to-point flying drone, its original plan represents a direct path from its current location to its destination. Highlighting the expected outcomes of this direct path not only explains why the drone had to reroute but also indicates when it can resume direct, point-to-point flight. Thus, this information element could more effectively reflect the current situation. Additionally, for consistency, using a cross symbol similar to those marking the expected outcomes of the new path may be better than red grid cells for indicating the expected outcomes of the direct path or original plan. The red grid cells rely on the shape of geofences, making them applicable only within the concept of Dynamic Airspace Reconfiguration (DAR). In contrast, the cross symbol is independent of geofences and can be applied to a broader range of operational concepts.

Since most engineering transparency elements were deemed relatively less useful in nominal scenarios, they could be simplified or condensed further to enhance their usability. For example, the search graph can be omitted for grid-based path-planning algorithms to reduce visual clutter. The explored nodes and search trees can be combined to indicate the space explored by an algorithm. As shown in Figure 4.5, the cost values have already been embedded within the explored nodes. In this way, only two transparency elements were required to reveal the inner workings of a path-planning algorithm: Explored Space and Search Process. The explored space represents the final results of the algorithm's exploration, while the search process illustrates the step-by-step details of how the exploration unfolds. They represent two distinct presentation styles for opening the "black box": a static image and a dynamic animation.

4.6.3. INTEGRATION OF TRANSPARENCY INTO UTM SYSTEMS

This research aligns with the U-space Concept of Operations (ConOps) [9], which mandates a collaborative interface with ATC to support human operators in managing drone traffic within controlled airspace. Several corresponding interface prototypes have been developed in previous studies [14–16, 63], with one of them presented in Figure 4.18. On the left side of the display is a selection panel where users can activate various options to display different information layers on the main radar screen (map). The proposed transparency elements can be integrated as additional options within this selection panel. As discussed in Section 4.6.2, the transparency elements can be further condensed, thereby limiting the number of options. Otherwise, presenting too many options may overwhelm users. The operational transparency elements can also be considered as contributing to the provision of certain UTM services, such as intent sharing, conflict alerts and weather information. All services can be integrated into a single interface, enabling UTM supervisors to share information seamlessly with drone operators.

In such a UTM system, a centralised (conflict-free path-planning) algorithm is expected to control all drones during tactical operations, meaning that flight plans are generated by the UTM system rather than by individual drones. This centralised approach

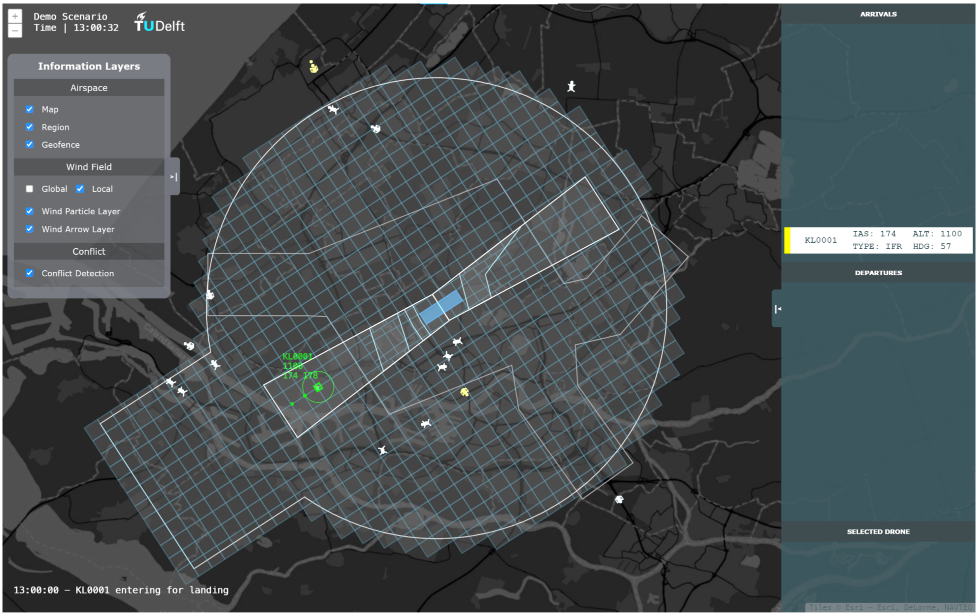


Figure 4.18: UTM interface prototype DroneCTR[1] developed by Janisch *et al.* [63].

allows both operational and engineering transparency elements to be computed directly within the UTM system, thereby reducing the volume of data that needs to be transferred between drones and UTM. If the UTM system is decentralised, the role of UTM supervisors may become unnecessary, and drone operators should be responsible for the safety of their own flights. An interface similar to Figure 4.18, integrated with the transparency elements, can be adapted to help drone operators monitor whether their drones are maintaining safe separation from other aircraft. As long as humans remain involved, transparency is essential for effective human-automation collaboration.

From a technical feasibility standpoint, integrating transparency elements into the UTM interface requires a robust backend architecture capable of real-time data processing and integration. The centralised conflict-free path-planning algorithm operates on a server-side platform, generating safe and efficient flight plans under dynamic airspace conditions. These flight plans should then be transmitted to the drones securely and reliably. To ensure accurate drone positioning, the system needs to continuously aggregate data from diverse sources, including Global Positioning System (GPS), onboard sensors, and external surveillance systems. It is also imperative to maintain low latency and support timely updates to the user interface.

4.6.4. TRAINING NEEDS FOR UTM SUPERVISION

As mentioned by both ATCos and drone experts, when scenarios become more complex (e.g., increased number of drones), they are more concerned about the risk of information overload, thereby preferring less transparency information. However, such traffic complexity also heightens the risk of conflicts between drones and crewed aircraft. Operators may in fact need greater transparency to support their supervision in complex scenarios. Moreover, when automation fails, which is often an urgent situation, ATCos and drone experts express a preference for access to nearly all available information. In such cases, the risk of information overload may be even more pronounced. Therefore, future UTM supervisors should be able to quickly identify the most relevant information in various situations. Appropriate training is required to ensure that they are familiar with how to effectively utilise transparency information to learn more about the nature of the supervisory control task and the system that needs to be monitored.

The proposed unified transparency taxonomy can also serve as a reference for training practices that rely on providing information "scaffolds" to guide the learning process in a phased manner, such as the Four-Component Instructional Design (4C/ID) model [203, 204]. The hierarchical structure of the proposed transparency taxonomy seems inline with progressively providing deeper (algorithmic) information, ranging from easily interpretable operational parameters to more complex engineering parameters.

For example, due to the large speed difference between drones and crewed aircraft, it may be difficult for inexperienced people to predict their Closest Point of Approach (CPA). In this case, the CPA-related transparency elements can provide valuable support. On the one hand, they can help operators learn how to predict CPA more accurately. On the other hand, they reassure operators that this information can be relied upon when they feel uncertain about their own predictions. Engineering transparency further helps operators understand an algorithm's capabilities and limitations, fostering well-calibrated trust, preventing over-reliance, and promoting learning – potentially reducing the need for transparency over time.

The hierarchical structure of the proposed taxonomy may also serve as procedural information to some extent, guiding operators through a step-by-step process for diagnosing issues such as automation failures. For instance, when UTM rerouting fails, the expected outcomes of the old path may help operators understand the direct cause, such as to avoid a conflict with a newly incoming crewed aircraft (a new constraint appears). Then, operators could further inspect the domain constraints, since the failure may have been caused by some other factors such as limited remaining battery or a strong headwind. Finally, they could examine the algorithm's inner workings to gain deeper insights. The grid size may be too large to find a feasible path, or the search tree may be overly constrained, preventing the drone from flying around certain obstacles. After repeated execution of this step-by-step process, operators could acquire sufficient understanding (of the algorithm and context) to reduce their reliance on transparency mechanisms.

4.6.5. LIMITATIONS AND FUTURE RESEARCH

In this chapter, the user study were conducted only via a questionnaire. Participants did not experience the actual functioning of transparency in (simulated) UTM operations, and thus their ratings were mainly based on their prior experience and expectations. The responses to the questionnaire can only provide subjective measurements that may be biased due to the small sample size. To address these limitations, future research (Chapter 5) will involve human-in-the-loop experiments in dynamic scenarios to further explore the practical usage of different transparency elements. During the second rating phase, participants were only required to rate elements in normal scenarios and the failure scenario may warrant further exploration.

Additionally, the engineering transparency categories in the unified taxonomy were largely derived from both literature and the experience with "traditional" path-planning algorithms such as graph-based and sampling-based algorithms. Traditional path planning, instead of machine learning-based path planning, was chosen for UTM routing because the operational UTM environment within controlled airspace is assumed to be fully known (similar to current ATC), and the future trajectories of all flights are generally predictable, making the traditional approach particularly suitable in this case. Traditional path planning can be a feasible and practical solution to UTM in the near future due to its solid mathematical foundations and theoretical guarantees. This "traditional" field also continues to evolve, with algorithms becoming increasingly faster and optimal [85].

However, owing to the substantial potential offered by machine learning, future research can explore how to extend this taxonomy to include machine learning methods as well, addressing transparency for training data, training algorithms and trained models [33]. For instance, visualising the policy in reinforcement learning could offer deeper insights into the AI decision-making strategy [205]. Revealing the learning process could assist policymakers in identifying bias in learning-based AI models [28].

4.7. CONCLUSION

This research introduces a unified taxonomy for algorithmic transparency, integrating established user-, ecology-, and model-centred perspectives to achieve operational, domain, and engineering transparency. Based on the taxonomy, twenty transparency elements and fourteen corresponding visual prototypes were designed to support the supervision of tactical UTM operations within CTR around airports. A survey-based user study was then conducted to investigate the needs and preferences of ATCos and drone experts on these elements in different scenarios. The results suggest that transparency is a dynamic construct that depends on situational demands and operator background. In nominal UTM scenarios, operational transparency is deemed more useful than engineering transparency. In the case of automation failure, operators tend to seek more engineering transparency to understand what happened deeper inside the system. The proposed transparency taxonomy offers the flexibility to accommodate these varying transparency needs across various scenarios. As scenarios become more complex, the issue of information overload may intensify. To mitigate this issue, appropriate training may be necessary for UTM supervisors to effectively access and interpret transparency information in different situations. The grouping results of the transparency elements validated the structure of the proposed taxonomy. As demonstrated in this chapter, the taxonomy could serve as a guide for system developers in designing transparency.

5

USAGE OF TRANSPARENCY FOR UTM SUPERVISION

In the previous chapter, various transparency elements were designed for supervising UTM, and a survey study was conducted to investigate operators' transparency needs. This chapter further conducts a human-in-the-loop experiment to explore how operators utilise transparency information in real-time UTM operations. It examines whether the information operators indicated a preference for in the survey aligns with what they actually use and need in practice.

The contents of this chapter are based on:

Paper title	Exploring the usage of transparency in supervising uncrewed air traffic management systems
Authors	Yiyuan Zou and Clark Borst
Submitted to	International Journal of Human-Computer Studies

ABSTRACT

To safely and efficiently manage the anticipated surge in drone traffic, Uncrewed Air Traffic Management (UTM) is currently being developed based on high levels of automation. In the low-altitude airspace surrounding airports, UTM aims to keep drones safely separated from crewed aircraft. However, automation deployed in safety-critical environments generally warrants human supervision to enhance overall safety and reliability. This requires some form of "seeing-into" transparency to provide humans with deeper insights into the automation. This chapter presents the design of a novel interface for UTM in Controlled Traffic Regions (CTR) around airports, which integrates eight distinct transparency information elements. A human-in-the-loop experiment, involving 16 participants, explored what and how specific transparency elements were used under different traffic conditions, algorithm settings and automation failures. The results reveal that information about the Closest Point of Approach (CPA) between drones and crewed aircraft is the most useful element for supporting UTM supervision. When UTM (re)routing fails, operators typically seek more information, such as constraint/situation changes and details about the algorithm's inner workings, to understand what happened and also gather clues for their intervention actions. As conflict-free path planning is central to UTM (re)routing, this research also compares the differences between a novel graph-based algorithm, Zeta-SIPP, and an advanced sampling-based algorithm, Informed RRT*. The findings suggest that the sampling-based algorithm might be more suitable for UTM supervision because it generally leads to a lower perceived workload and its search tree visualisation could better support human interventions in addressing automation failures.*

5.1. INTRODUCTION

In recent years, the use of drones in various domains, such as delivery, inspection, aerial mapping, and rescue, has substantially increased due to their unique benefits [206]. By the end of 2023, over 842,000 small commercial drones (weighing between 250 grams and 25 kg) had been registered in the United States [2]. The Federal Aviation Administration (FAA) forecast that the commercial drone fleet would likely increase to around 1.12 million by 2028 [2]. Due to the small size, drones will operate at much higher densities than crewed aircraft [207] and are more susceptible to wind [208], introducing many new challenges to traditional Air Traffic Management (ATM) methods. To safely and efficiently manage the large number of drones, Uncrewed Air Traffic Management (UTM) is currently under development around the world [6, 8, 9, 173]. In Europe, SESAR 3 Joint Undertaking launched multiple projects to explore various issues of U-space, the European version of UTM, such as CORUS five (fifth version of U-space concept of operations) [10], ENSURE (ATM–UTM collaboration) [209] and MUSE (societal and environmental impact) [210].

Due to the expected high volume of drone operations, UTM will be built based on high levels of automation [9]. As 100% safe and reliable automation does not exist yet, UTM still requires human supervision, particularly in low-altitude airspace around airports, where the risk of collisions between drones and crewed aircraft remains a concern. However, a higher level of automation usually makes it more difficult for humans to understand the rationale behind automation behaviour, potentially reducing their situation awareness, trust and acceptance [111, 169]. Operator acceptance has proven to be one of the largest obstacles to successfully introducing new advanced technologies to aviation [110, 112, 113]. Therefore, some form of "seeing-into" transparency that reveals the inner workings of automation may be required to enhance human understanding, address trust and acceptance issues, and support human supervision [20, 22, 70, 176, 211].

As UTM is a novel concept, direct research on transparent UTM is still limited [14, 62]. Some research focused more on human-machine interactions in UTM or (one-to-many) drone operations [59, 178, 191, 198, 212, 213] while others mainly addressed multi-unmanned (air, ground, and surface) vehicle control [23–25, 171]. Their results generally indicated positive effects of transparency on human performance [23–25, 178]. However, these studies primarily concerned mission planning rather than traffic management like ATM and UTM. The core task of UTM is to guide drones to their destinations while avoiding any conflicts or collisions. This research focuses mainly on tactical UTM operations around airports, where the key challenge is ensuring safe separation between drones and crewed aircraft in real time [14, 16]. The impact and usage of transparency in this case may thus be different, which are still unclear and underexplored.

Although transparency research in UTM is limited, some progress has been made in the ATM field. SESAR 3 Joint Undertaking initiated five projects that addressed the transparency and explainability issues of automation in ATM [36]. Explainable AI (XAI) has been introduced to enhance the transparency of AI-based ATM systems [180, 181, 183]. However, as indicated by [62], research on transparency usually dealt with similar problems using different yet related approaches, such as the ARTIMATION [60], MAHALO [61] and TAPAS [55] projects under SESAR [36], lacking a clear and unified design guidance.

To address this issue, a unified transparency taxonomy was proposed in Chapter 4, as presented in Figure 5.1. Three concepts of transparency have been introduced: *operational*, *engineering* and *domain* transparency. Operational transparency reveals information related to the operational situations and system states, aiming to help operators maintain situation awareness. In contrast, engineering transparency provides insights into the inner workings of models and/or algorithms, fostering human understanding, trust, and collaboration with them. Domain transparency, found in the middle, emphasises the disclosure of the physical and intentional constraints that govern the work domain, independent of the agent that performs actions. It creates a shared foundation for all agents, as both humans and machines must adhere to the same domain constraints. Based on the proposed taxonomy, a survey-based user study was conducted to investigate the transparency needs of Air Traffic Controllers (ATCos) and drone operators if they were to supervise UTM [62]. The findings suggest that operational transparency is preferable to engineering transparency in UTM, with no significant difference observed between ATCos and drone operators.

This research extends previous studies [14, 62, 63], further exploring *when* and *how* operators would use transparency in tactical UTM operations via a human-in-the-loop experiment. Unlike the previous survey-based user study in Chapter 4, which featured only static scenarios, participants in this experiment interact with various dynamic environments, including nominal operations, automation failures, and dense traffic conditions. Through direct experience with tactical UTM operations, operators' opinions and preferences regarding the usefulness of certain transparency information may differ

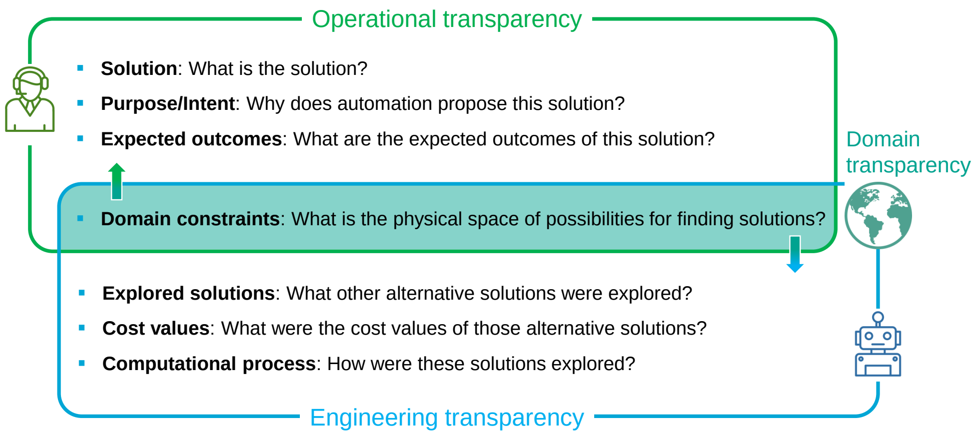


Figure 5.1: Proposed unified taxonomy for algorithmic transparency.

from those reported in the survey study. Furthermore, tactical UTM operations are time-constrained and safety-critical. Varying levels of time pressure and workload may influence the need for and usage of transparency. A comparison is conducted between a novel graph-based path-planning algorithm, Zeta*-SIPP [85], and an advanced sampling-based algorithm, Informed RRT* [83], for supervising UTM operations. This choice is motivated by the study in Chapter 3, which indicates that algorithm type can affect human understanding [214]. Overall, this research contributes to the development of interactive and transparent path planning in real-time, safety-critical contexts.

This chapter is structured as follows: Section 5.2 provides a literature review of transparency frameworks, empirical studies and transparent ATM/UTM. Section 5.3 shows the developed UTM interface prototype, incorporating various transparency elements. Section 5.4 outlines the experiment design, including the setup, participant recruitment, independent variables, dependent measures, control variables and hypotheses. Section 5.5 presents the results regarding safety, efficiency, transparency usage and preference, intervention, workload and interface acceptance. Finally, Section 5.6 summarises the study findings, discusses their implications, and highlights future research directions.

5.2. RELATED WORK

5.2.1. DESIGN FRAMEWORKS

To achieve automation transparency, several guidelines and frameworks have been proposed for designers. For example, inspired by folk psychology, the Belief-Desire-Intention (BDI) framework was introduced [132] and has been applied to generate explanations of agent behaviour [215, 216]. Research suggested that explanations should be designed by combining both goals (desires) and beliefs [216–218]. For human trust in automation, Lee and See [133] proposed the 3Ps theory: Purpose, Process and Performance. They indicated that automation should disclose why it was developed (purpose), how it operates (process) and what it does (performance). To support human-robot interactions, Lyons [21] developed a human-robot transparency model that includes robot-to-human and robot-of-human transparency. For robot-to-human transparency, four types of information should be presented to humans: 1) the robot's design, purpose and intent (intentional model), 2) its tasks, goals and progresses (task model), 3) its underlying analytical structure (analytical model), and 4) its understanding of the environmental conditions (environmental model). For robot-of-human transparency, both humans and robots should have a shared understanding of each other's roles (teamwork model) and robots should also communicate their understanding of humans' cognitive, emotional, and physical states (human state model). Inspired by Endsley's [135] theory of Situation Awareness (SA), the BDI framework [132] and Lee and See's 3Ps theory [133], Chen et al. [22] developed a SA-based Agent Transparency (SAT) model. To support the three levels of SA, the agent's basic information (Level 1: current status, intentions and plans), rationale (Level 2: reasoning process) and outcomes (Level 3: projections and limitations) should be revealed.

In recent years, the rapid development of AI has increasingly attracted the attention of researchers to its explainability [32, 33]. Explainable AI (XAI) is an emerging field, referring to "AI systems that can explain their rationale to a human user, characterise their strengths and weaknesses, and convey an understanding of how they will behave in the future" [31]. Unlike previous research on agent transparency [21, 22], XAI puts more emphasis on explaining the inner workings of advanced AI technology, aiming to open the "black-box" to achieve model or algorithmic transparency [26, 30, 34]. Several frameworks have been proposed for designing XAI systems. For example, inspired by the theoretical underpinnings of human decision making, Wang et al. [131] proposed a theory-driven user-centric XAI framework to support human reasoning processes and reduce cognitive biases. Eiband et al. [130] introduced a stage-based participatory process for designing transparent interfaces. The first three stages aim to identify the content of explanations (what to explain), involving what can be explained from expert perspectives, the differences between the user and expert mental models, and what users want and prefer. The last two stages focus on determining the presentation format (how to explain), including iterative prototyping and evaluation. Building upon these foundations, Mohseni et al. [114] further developed a nested XAI framework with three layers. The outer layer defines the goals of the XAI system, the middle layer designs the interface to meet user needs, and the innermost layer explains the underlying algorithms. In each layer, design and evaluation form an iterative cycle. Furthermore, Kaplan et al. [219] proposed a unified XAI framework by addressing four key questions: 1) why explain, 2) what to explain, 3) for whom to explain, and 4) how to explain. The framework comprises five aspects: system, data, model, performance, and decision exploration, aiming to provide explanations tailored to different user groups.

The guidelines and frameworks mentioned above mainly focus on revealing information about agents or robots themselves, while somewhat overlooking the disclosure of domain constraints. Although Lyons' human-robot transparency model and the SAT model both indicate environmental constraints, these constraints mainly refer to those perceived by agents, which may be inaccurate if the agents rely only on their own sensory information. In fact, domain constraints should be independent of agents because all (human and machine) agents need to obey the same domain constraints. To uncover the underlying structure of work domains, Ecological Interface Design (EID) [53, 153] and Cognitive Work Analysis (CWA) [57] were developed. The ecological approach aims to visualise the physical and intentional constraints that govern the work domain, intuitively revealing the deep structure to facilitate *domain* transparency [107, 108, 187, 220]. EID is guided by three principles, each designed to support a different level of cognitive control: skill-, rule- and knowledge-based behaviours (SRK taxonomy) [221]. Recent research attempted to integrate Lyons' human-robot transparency model and the SAT model into EID [222] and introduce CWA to XAI [58]. A user study has shown that revealing domain knowledge could prevent over-reliance on automation and foster appropriate levels of trust [150].

In summary, transparency research can generally be divided into three distinct perspectives: user-centred approaches, model-centred approaches and ecology-centred approaches [62]. The model-centred approaches seek to reveal relevant information about (internal) models, with less focus on the actual (external) context or situation in which the information is used. The user-centred approaches put more emphasis on user demands and experience, making explanations and disclosed information align more closely with user preferences, beliefs and sensory capabilities. The ecology-centred approaches revolve around the work domain and environment, seeking to make complex domain constraints salient. As illustrated in Figure 5.1, these perspectives can be integrated into a unified transparency taxonomy, structured around the *solution* generated by automation in a supervisory control setting. The proposed operational transparency categories can be regarded as a variant of the SAT model, while the engineering transparency reveals the internal processes of algorithms governing automation behaviour. The domain transparency is positioned at the intersection, serving both to explain the feasibility and robustness of solutions (operational) and to clarify the boundary of automation optimisation (engineering). According to the proposed transparency taxonomy, a novel interface for UTM supervision was developed, integrating various transparency elements. More details will be discussed in Section 5.3.

5

5.2.2. EMPIRICAL EVIDENCE

As indicated by Langer *et al.* [111], improving human understanding is a crucial objective of transparency and plays a central role in satisfying the desires of stakeholders. As the level of transparency increases, humans can generally form more accurate mental models of the underlying system's functioning [30, 34, 142, 145], which often leads to improvements in their trust and acceptance [139, 223, 224]. However, some studies reported mixed results [30, 138, 143, 144], identifying that transparency does not always result in positive effects. There are several factors that influence the effectiveness of transparency. In many cases, the negative impact of transparency can be attributed to *expectation violation*, where the outputs and/or actual workings of automation violate users' initial expectations [30, 147, 149]. An appropriate expectation adjustment may be necessary for increasing user satisfaction and acceptance [148]. When receiving more information, humans tend to be more confident regardless of the accuracy of their judgments [165] and are more easily biased towards agreeing with automation [25, 225]. The effectiveness of explanations also depends on their meaningfulness to humans [146]. The disclosed information and explanations should be presented in a human-understandable manner, utilising various modalities such as textual, verbal, and visual transparency [145, 226].

The effects of transparency on human-machine teaming in tactical operations have also been investigated. Many studies have identified that increased transparency benefits operators' situation awareness and performance without increasing – and in some cases even reducing – workload [23, 24, 178, 227–229]. However, reports on operators' decision time were mixed. Some studies suggested that increased transparency reduces or has no impact on decision time [23, 178, 227, 230], while some revealed that decision time may increase due to the additional time for processing more information [24, 229]. Time pressure is an important issue in tactical operations, which may affect the usage of transparency [25, 228, 229]. In the experiment conducted by Hurter et al. [60], ATCos preferred little transparency due to the high time pressure.

This study primarily explores the usage of transparency in tactical UTM operations around airports. In this context, operators need to continuously monitor the automated UTM system to prevent conflicts between drones and crewed aircraft, and promptly intervene when necessary. Due to the limited time available to resolve issues, such as UTM failures in deconfliction, rapid and effective responses are critical. Since previous studies have demonstrated the benefits of transparency, this research focuses more on *what*, *how* and *when* transparency information would be used in such real-time, safety-critical operations via a human-in-the-loop experiment.

5

5.2.3. Transparency in UTM and ATM

As UTM has not been fully established yet, research on transparent UTM remains limited. Some works attempted to design UTM interfaces or systems [14, 15, 231–233] while some explored UTM operational concepts that support human-machine interactions [59, 177, 212]. The study in Chapter 4 investigated the transparency needs of professional ATCos and drone operators for supervising UTM systems. The results indicated that both groups preferred operational transparency over engineering transparency, and their needs varied across different scenarios. Almost everyone agreed that transparency is important for UTM and would significantly influence their level of acceptance and trust.

Given the similarities between UTM and ATM, the ATM transparency research is also reviewed, such as the ARTIMATION [60], MAHALO [61] and TAPAS [55] projects under SESAR. ARTIMATION introduced three levels of transparency: 1) Black Box, 2) Heat Map, and 3) Storytelling. The Black Box displayed only the proposed solution along with execution instructions, without revealing the underlying process. The Heat Map visualised the algorithm's explored trajectories, indicating whether they were safe or not. The Storytelling offered a step-by-step preview of the proposed solution while also presenting alternative options. MAHALO proposed three transparency conditions: 1) Vector Line, 2) Vector Line and Solution Space Diagram (SSD), and 3) Vector Line, SSD and text-based explanation. The Vector Line, indicating aircraft speed and heading, represented the proposed solution for conflict resolution. The text-based explanation clarified the target Closest Point of Approach (CPA) and the agent's purpose and intent. The foundation of MAHALO is SSD, which could *visually* illustrate the feasibility and robustness of the proposed solution. TAPAS primarily implemented visual panels and textual tables to present detailed information about detected conflicts, such as conflict geometry and severity, along with suggested solutions and their expected outcomes. Their transparency design is based on Endsley's situation awareness theory [135], similar to the SAT model. They also introduced the concept of operational transparency but did not explore domain and engineering transparency. Additionally, there are several other works that aim to enhance transparency in AI-based ATM systems by applying XAI methods [180, 181, 183], such as Local Interpretable Model-agnostic Explanations (LIME) [93] and SHapley Additive exPlanations (SHAP) [94].

In summary, the previous works lead to different solutions for ATM transparency. For example, the three SESAR projects ARTIMATION, MAHALO, and TAPAS all addressed the same problem in ATM (i.e., conflict detection and resolution), but arrived at different design choices. ARTIMATION focused on engineering transparency by revealing the algorithm's inner workings, such as heat maps that visualise the trajectories explored by the algorithm. In contrast, MAHALO emphasised domain transparency, using SSD to represent all available solutions. SSD not only demonstrates the feasibility of the proposed solution (i.e., Vector Line) but also enables operators to manually select their preferred alternative. TAPAS, meanwhile, concentrated on operational transparency and explicitly introduced this concept within its research.

Therefore, a unified taxonomy covering different aspects of transparency was devised in Chapter 4, as shown in Figure 5.1, and accordingly various transparency elements were designed for supervising UTM. This chapter will present an interactive interface prototype that visually portrays these elements. The human-in-the-loop experiment in this research aims to explore how these elements would be utilised for supervising UTM routing decisions and maintaining safe separation between drones and crewed aircraft.

5.3. INTERFACE DESIGN

5.3.1. INTERFACE PROTOTYPE

Given the broad scope of UTM, this research revolves around tactical UTM operations within Controlled Traffic Regions (CTR) near airports [14, 63], in particular Rotterdam The Hague Airport. Due to the large range of CTR (e.g., 8 nmi around Rotterdam The Hague Airport [12]), some urban airspace, such as Amsterdam and Rotterdam, is mostly covered by them, greatly hindering drone operations in cities. In recent years, unannounced private drone operations at Gatwick, Heathrow, Frankfurt and Madrid caused large disruptions in regular air traffic flows, posing huge risks to crewed aviation [4].

To support UTM in the CTR, an interface prototype, called DroneCTR [14, 63], was developed and enhanced with various transparency elements [62], as shown in Figure 5.2. The transparency elements proposed in the previous work are implemented as the eight buttons on the left side of the interface. Users can activate a certain element by pressing the corresponding button. When an element is deactivated, it will be removed from the map. This is an *adaptable* control approach that enables users to drive changes in the interface [171]. It helps operators maintain situation awareness and prevents "automation surprises" caused by unexpected automation-driven changes (adaptive automation) [171, 234]. More details of the implemented transparency elements will be introduced in the next section.

On the map, the circle surrounding a crewed aircraft represents its safe separation from drones, which is assumed as 1 km in this research [14, 63]. Given that the safe sep-

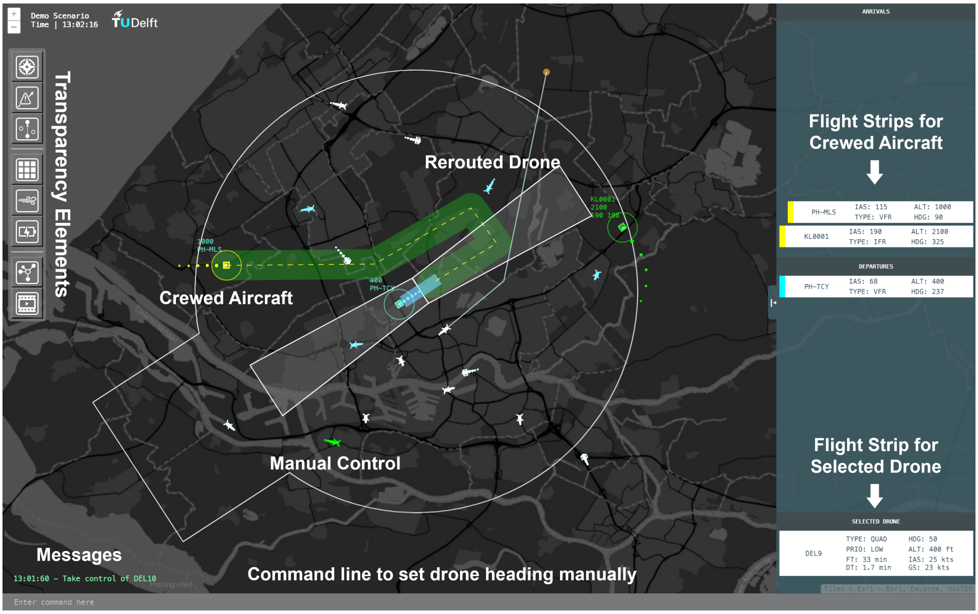


Figure 5.2: UTM interface prototype DroneCTR[1]. To control variables, all drones perform point-to-point delivery missions in this research.

aration is a crucial criterion for UTM, the safe separation circles are always displayed in the interface. Users can view aircraft routes by clicking on aircraft icons. When a crewed aircraft is clicked, a green air corridor is presented, which is associated with the aircraft route and safe separation. As aircraft routes are basic elements for operators to interpret aircraft intents, this information element is not allowed to be removed using a button like the one on the left. Similar to traditional ATM displays, history dots trailing the aircraft icon are presented to indicate the aircraft locations during the past few dozen seconds. The right side of the interface displays the flight strips of crewed aircraft and drones, providing detailed information about each flight. The panel containing the flight strips can be shown or hidden using the button attached to its side.

The lower-left corner of the interface shows event messages. When a certain event occurs, a message will pop up as an indicator for around 35 seconds. There are five types of messages with different colours: changes to air traffic (white), automation-triggered events (blue), automation failures (orange with a box border), manual control actions (green), and loss of separation alerts (red with a box border). To avoid message overflow, where multiple messages appear simultaneously and occupy a large portion of the interface, the display is limited to a maximum of six messages at a time. An example of event messages is shown in Figure 5.3. The grey background of the message signifies that it is being hovered over by the mouse cursor, while the cursor changing to a hand indicates that the element is clickable. Operators can click on a message to select the corresponding drone or crewed aircraft on the map.

[1] URL: `http://dronectr.tudelft.nl/`, ID: transparency

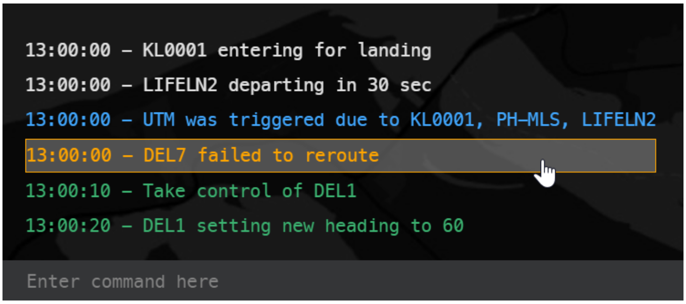


Figure 5.3: An example of event messages.

The concept of operations for UTM within CTR has been explored in several previous works [14–16, 63, 177]. They followed the U-space framework [9] and implemented a centralised UTM system, as depicted in Figure 5.4. In this case, UTM provides operators with information to monitor the automated routing service, while also allowing them to input instructions or constraints to intervene when necessary. The interface prototype also adopted this centralised approach. Geofences, designated as no-fly zones for drones, have been proposed to achieve collaborative operations between ATM and UTM. Operators can dynamically add or remove geofences to prevent drones from entering certain areas, thus separating drones from crewed aircraft. This concept is called Dynamic Airspace Reconfiguration (DAR). This approach eliminates the need for operators to manually control each drone, making it ideal for managing a large number of drones. However, previous experiments [14, 15, 63] showed that operators also expressed a preference for active control over specific drones instead of only dynamic geofencing (i.e., passive control). Therefore, this research will include more possibilities for active control. A new feature, Command Line, has been implemented to enable operators to issue direct commands to drones, as shown at the bottom of the interface. For example, one can input "DEL9 HDG 90" and press the "Enter" key to set the heading of the drone DEL9 to 90°. In this case, DEL9 is no longer under UTM control and its icon colour turns green. For reasons of experimental control, operators were not allowed to use the Command Line to change the drone speed and altitude in this research.

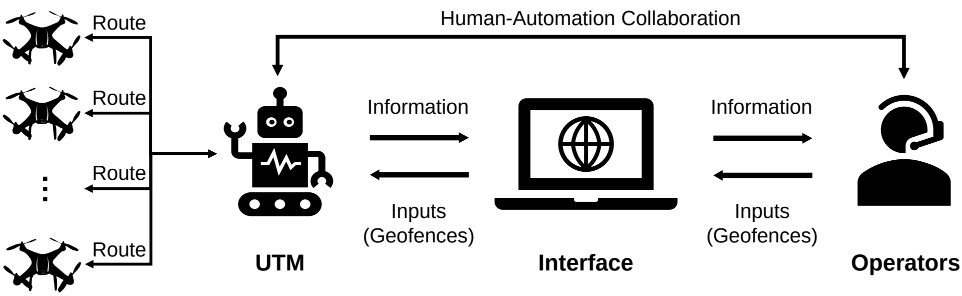
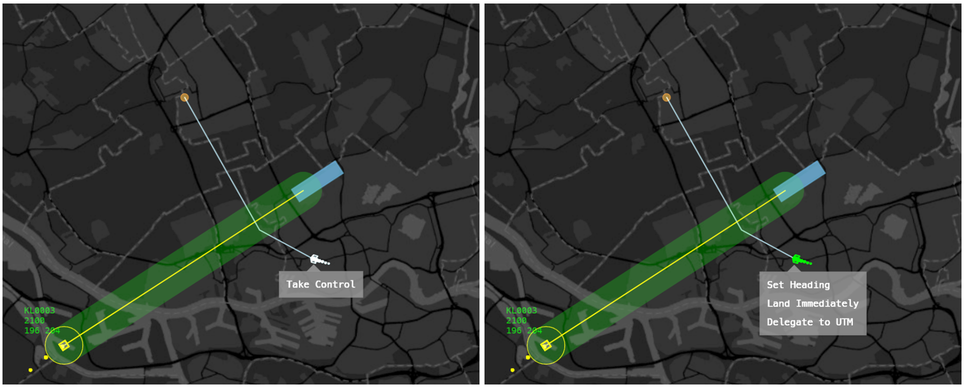


Figure 5.4: Centralised UTM for drone (re-)routing, which was also adopted in the interface prototype.

This concept of operations is also similar to the flight-based control allocation [235]. All drones under UTM control are coloured blue, but operators can take control over a specific drone from UTM when necessary, changing the drone's colour to green. When a drone is under human control, operators have full responsibility for ensuring its safety and successful mission completion. The geofences (also coloured in blue) are only applicable to the blue drones whereas the green drones will ignore them. When a drone is rerouted by UTM, its icon changes to a darker blue as a visual indicator. To clarify the division of responsibilities, operators need to first take control before they can set a drone's heading. As shown in Figure 5.5, users can right-click on the drone icon to open the control menu. After taking control, operators can select one of the available commands for the drone. The "Land Immediately" command is used when operators believe the drone cannot avoid conflicts with crewed aircraft or complete its mission. They can also return control of the drone to UTM using the "Delegate to UTM" command. Also, when a drone is under human control, operators can right-click on its destination to direct the drone to fly there.



(a) Take control (b) Control commands

Figure 5.5: Exemplary screenshots of the control commands.

In this research, two advanced path-planning algorithms have been implemented for UTM routing: Zeta*-SIPP [85] and Informed RRT* [83]. This choice is motivated by the previous study in Chapter 3, which also selected a graph-based and a sampling-based algorithm, demonstrating that the type of algorithm can influence human understanding. However, different from the previous study, the algorithms applied in this research consider not only static obstacles (geofences) but also dynamic obstacles (crewed aircraft). Zeta*-SIPP is a state-of-the-art grid-based algorithm that can find time-optimal conflict-free paths with relatively less turns (any-angle path planning). Zeta*-SIPP first examines whether the direct path from the start point to the target point is conflict-free. If not, it gradually expands its search range in an elliptical pattern until the "current" optimal path becomes feasible. For simplicity, the "wait" action is disabled for path planning in this research. Informed RRT* is an advanced sampling-based algorithm that randomly expands its search tree to explore the search space. It can quickly find a feasible path and then continuously improves it until it converges to the optimal solution. The original Informed RRT* was designed for static environments, but in this research, it was adapted to deal with dynamic obstacles as well. At each iteration, when Informed RRT* attempts to add a new node to its search tree, it performs a line-of-sight check to ensure that the branch connecting the existing tree to the new node is not blocked by any static obstacles. To handle dynamic obstacles, an additional conflict detection check was introduced. A new branch is inserted into the search tree only if it is confirmed to be conflict-free.

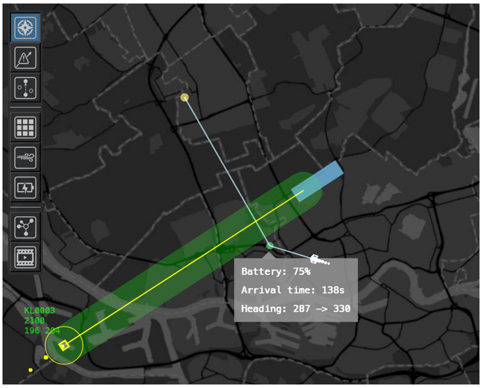
In summary, Zeta*-SIPP checks whether the current optimal path is feasible while Informed RRT* seeks to optimise the current feasible path. They represent two different directions for optimal path planning. Moreover, Zeta*-SIPP and Informed RRT* both incorporate an elliptical search/sampling range. Zeta*-SIPP incrementally expands its elliptical search range outward from the direct line connecting the start and target points until it either finds the optimal path or exhausts the search space. The search range represents the currently identified lower bound of the optimal path cost. In contrast, Informed RRT* progressively narrows its elliptical sampling range within the search space as better paths are discovered. The sampling range defines the upper bound for its random sampling process. The connections and differences between these two algorithms prompted a comparison in the experiment. Given that sampling-based algorithms may be easier to comprehend than graph-based algorithms [214], this study further investigates whether this trend also holds in tactical UTM supervision involving dynamic scenarios.

5.3.2. TRANSPARENCY ELEMENTS

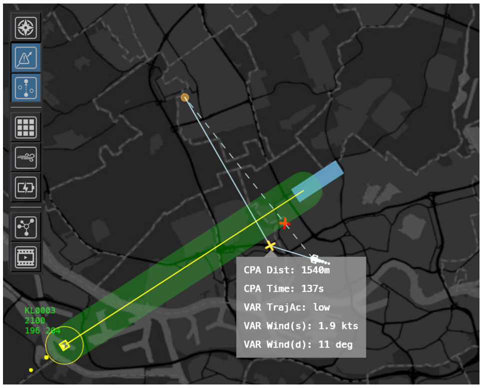
This section introduces the eight transparency elements (buttons) in the interface in details. In Chapter 4, twenty transparency elements were proposed for supporting UTM supervision based on the taxonomy in Figure 5.1. A survey study then revealed that operators prefer operational transparency over engineering transparency. They generally wanted their interface to be clean and not overwhelming. Moreover, another user study in Chapter 3 concluded that certain transparency elements regarding the inner workings of path-planning algorithms could be merged, as their individual contributions to human understanding or confidence were not substantial. Therefore, this research further condenses the relevant transparency information, integrating the twenty elements into eight ones, as shown in Figures 5.6 and 5.7. To reduce the text space occupied by images, the displays of Elements 2 and 3, as well as Elements 4 and 5, are combined. Corresponding to the transparency taxonomy, Elements 1-3 provide operational transparency, Elements 4-6 offer domain transparency, and Elements 7-8 reveal engineering transparency.

The first element is about waypoint states. As shown in Figure 5.6a, when users activate this element, green dots will be presented at the waypoints of drone routes. Users can right-click on a green dot to access detailed information about this waypoint, such as remaining battery, estimated arrival time and heading change. Elements 2 and 3 pertain to conflict detection. When Element 2 is activated, a cross symbol will appear at a certain location along the drone's flying route, indicating the Closest Point of Approach (CPA) between the drone and other crewed aircraft. Similarly, activating Element 3 will display a cross symbol along the drone's direct route, representing the CPA if the drone flew directly to its destination. The default colour of CPA crosses is orange, but the cross turns red if the separation distance at CPA is less than the safe separation threshold (1 km). Users can right-click on a cross to view more information about CPA, as shown in

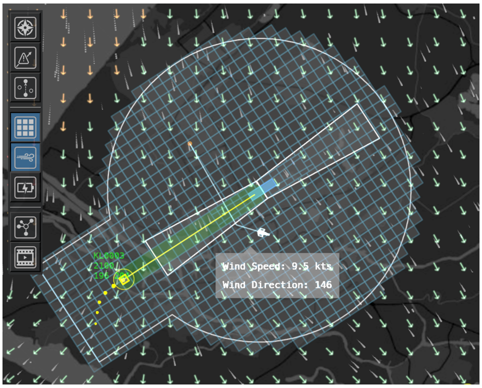
5



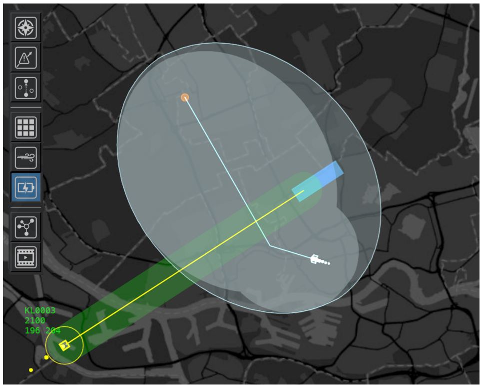
(a) Waypoint state (Element 1)



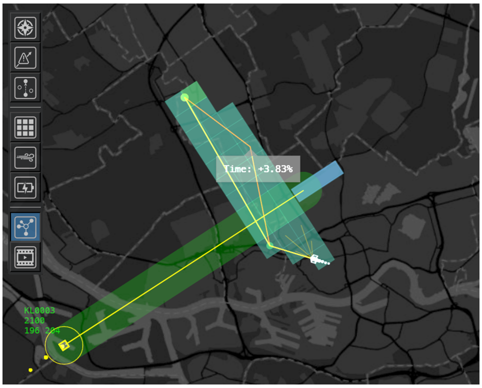
(b) Detection: flying and direct routes (Elements 2 and 3)



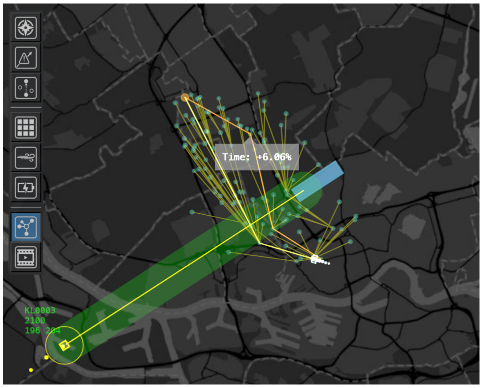
(c) Geofence grids and wind field (Elements 4 and 5)



(d) Drone manoeuvring space (Element 6)

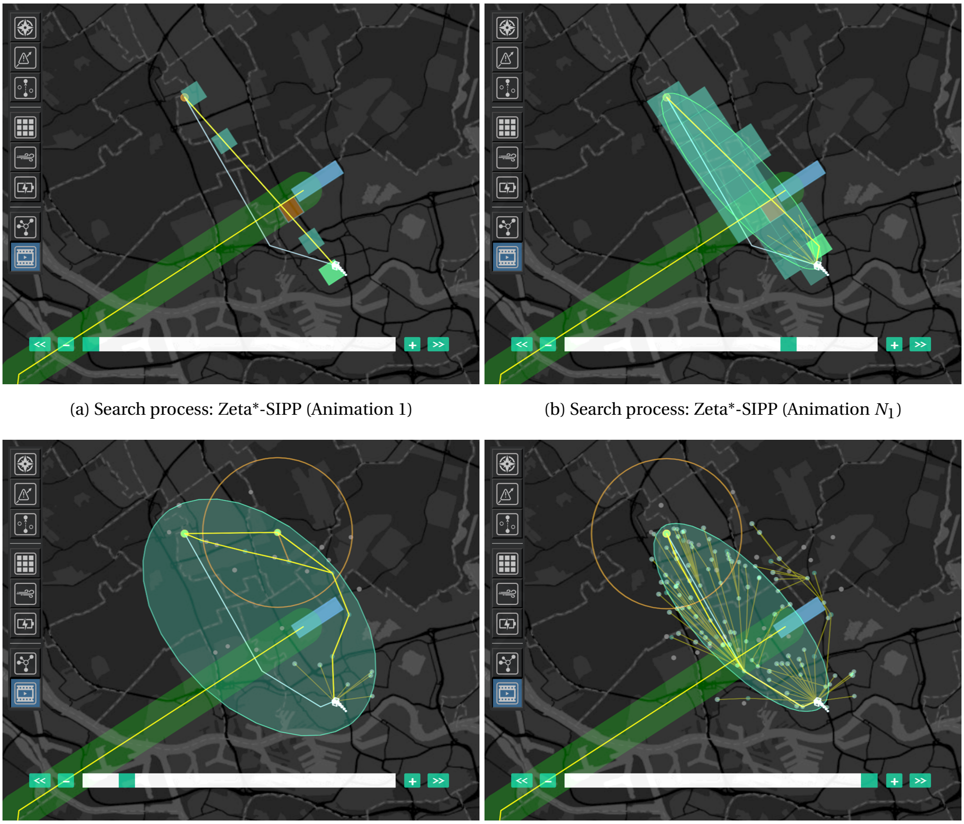


(e) Explored space: Zeta*-SIPP (Element 7)



(f) Explored space: Informed RRT* (Element 7)

Figure 5.6: Exemplary screenshots of different transparency elements.



(a) Search process: Zeta*-SIPP (Animation 1)

(b) Search process: Zeta*-SIPP (Animation N_1)

(c) Search process: Informed RRT* (Animation 1)

(d) Search process: Informed RRT* (Animation N_2)

Figure 5.7: Exemplary screenshots of search processes (Element 8).

Figure 5.6b. The “CPA Dist” refers to the separation distance at CPA and the “CPA Time” represents the predicted time to reach CPA.

However, due to the susceptibility of drones to wind and the dynamic trajectories of crewed aircraft, there are uncertainties in the CPA predictions. In the interface, the standard deviations of the local wind field around a drone are disclosed [236]: angular dispersion around the average wind vector (“VAR Wind (d)”) and wind speed dispersion along the average wind vector (“VAR Wind (s)”). For the trajectory uncertainty (“VAR TrajAc”), if the crewed aircraft associated with the drone’s CPA follows Visual Flight Rules (VFR), the uncertainty level is set to “med” (medium) considering that VFR flights are generally more flexible. In contrast, Instrument Flight Rules (IFR) aircraft and emergency helicopters are regarded as having “low” uncertainty since they typically follow structured or direct routes with fewer turns and exhibit more predictable behaviour.

Element 4 denotes geofence grids, representing a constraint for users to define geofences. Users cannot flexibly change their shape and size. Element 5 denotes the wind field, shown in the form of animated particles and coloured arrows. Different colours indicate different wind speeds: green denotes winds that have relatively little impact on drones while red represents strong winds that could cause drones to drift. One can also right-click on a wind arrow to view the wind speed and direction at a specific location. Element 6 shows the drone manoeuvring space [14]. It is a visual representation of the drone’s range governed by battery power and environmental conditions such as wind. Generally, a narrower manoeuvring space indicates lower excess battery power and/or increased headwind conditions. The manoeuvring spaces for both the flying and direct routes are presented. The outer manoeuvring space also defines the search space for path-planning algorithms. As a drone flying beyond this boundary is unable to return to its destination due to limited battery capacity, there is no need to search outside this boundary.

In Chapter 4, five transparency elements were devised for path-planning algorithms: search graphs, explored nodes, search trees, cost values and search processes. However, the survey study indicated that operators have limited need for these elements in nominal scenarios. Therefore, these elements are integrated into two ones in this research: a static image representing the entire explored space (Element 7) and a set of browsable images depicting the dynamic search process (Element 8). Element 7 illustrates the space explored by the algorithm, comprising both explored nodes and search trees. To reduce visual clutter, the search graph is excluded, and the cost values are embedded within the nodes. Users can right click on a node to retrieve the cost value of an explored path passing through that node (orange line). The cost value is presented as the ratio of the retrieved explored path to the current optimal path. As depicted in Figure 5.7, Element 8 provides users with access to the search process of the path-planning algorithm. An interactive slider is implemented, allowing users to progressively view the entire search process in either forward or backward directions.

As mentioned in the previous section, Zeta*-SIPP starts from the direct route, searches with an expanding elliptical range and stops when the current optimal path is feasible (no red cells block the path). In contrast, Informed RRT* randomly explores the search space, narrows it down upon finding feasible paths, and continues until a specified termination condition is met: either 200 samples are drawn or 150 tree nodes are reached. This condition is intentionally selected to ensure that Informed RRT* fails in cases where Zeta*-SIPP is also unable to find a path during the experiment. This helps ensure the scenarios for both algorithms remain as consistent as possible. The orange circle indicates the rewiring range of Informed RRT*, defined as $d_{st}/3$ [214], where d_{st} is the distance between the start and target points. The incremental distance for node expansion is set to $d_{st}/5$ [214].

5.4. METHODOLOGY

The previous section introduced the UTM interface prototype and its associated transparency elements. This section will present a human-in-the-loop experiment to explore how and when these proposed transparency elements are utilised by operators in supervising UTM operations.

5.4.1. EXPERIMENT SETUP

The experiment was conducted in the ATM laboratory at TU Delft based on the developed UTM interface, as shown in Figure 5.8. Each participant required around 2.5 to 3 hours to complete the experiment, which consisted of an interactive briefing session, a training round, a measurement round and a post-hoc questionnaire. During the briefing session, participants were introduced to the UTM background, the basic interface elements and the available control actions. There were 10 training runs and 8 measurement runs in total. Each run, except for the first two training runs, lasted 5 minutes. After each measurement run, participants were required to answer several questions to assess their workload and evaluate how effectively the interface supported their task completion in the scenario.

Figure 5.8: Experiment setup in the laboratory

In the experiment, all drones performed point-to-point delivery missions. The task of participants was to avoid loss of (horizontal) separation between drones and crewed aircraft (1 km) while ensuring that the drones completed their missions with minimal delays. The vertical separation was not considered. Participants could only influence the routes of drones, with crewed aircraft considered dynamic obstacles. By default, all drones were controlled by UTM using a path-planning algorithm, but participants were allowed to take control over specific drones from UTM. The path-planning algorithm was automatically triggered when a new crewed aircraft was about to enter the airspace (arrival or departure). In this case, a new message will pop up as an indicator (e.g., Figure 5.3) and the rerouted drones will change from light blue to darker blue.

5.4.2. PARTICIPANTS

Sixteen participants, all TU Delft staff and students, volunteered to take part in the experiment in the ATM laboratory. Their ages were distributed as follows: three participants were between 18 and 24, twelve between 25 and 34, and one between 55 and 64. All of them had a relevant background in aerospace engineering and drone operations. They were either drone experts or UTM researchers. Previous studies [15, 16, 63] suggested that the role of UTM supervisor cannot be assigned to tower controllers for workload reasons. Therefore, it has been suggested that a new position is needed that solely focuses on UTM supervision. This position does not have to be a professional ATCo and should ideally be a person who is more familiar with drone behaviour and performances. The large speed difference between crewed aircraft and drones makes it difficult for ATCos to perceive the urgency of conflicts and when to take actions [63] as their training only focused on handling crewed aircraft. The previous survey study in Chapter 4 also revealed no significant differences in transparency needs between ATCos and drone operators. Therefore, drone experts and UTM researchers were recruited for the experiment rather than professional ATCos. This experiment was approved by the Human Research Ethics Committee (HREC) under number 4724.

5.4.3. INDEPENDENT VARIABLES

The experiment had two within-participants independent variables: the path-planning algorithm, having two levels and the traffic scenario, having four levels.

Path-planning algorithms. As mentioned in the previous section, Zeta*-SIPP and Informed RRT* were chosen for the experiment. This was because they are advanced examples of graph and sampling-based algorithms, representing distinct approaches to path planning with dynamic obstacles. The relevant engineering transparency elements are also different based on the algorithms used. Investigating the influence of the algorithm type on transparency usage and identifying which algorithm is more suitable for UTM supervision could offer valuable insights for system designers.

Traffic scenarios. The previous survey study in Chapter 4 found that the type of traffic scenarios affects the transparency needs of operators for supervising UTM systems. Thus, this experiment further explored how different traffic scenarios influence the transparency usage in real-time dynamic operations. Four traffic scenarios were then defined: simple-normal, simple-failure, complex-normal and complex-failure. The terms "simple" and "complex" mainly describe the complexity of the scenarios, which is determined not only by the number of drones and crewed aircraft, but also by the diversity of their flight types and the number of interactions between them. The "normal" and "failure" indicate whether an automation failure event occurs (i.e., the path-planning algorithm fails to find a feasible path). Table 5.1 presents the characteristics of different traffic scenarios. This design was also informed by previous work on UTM operations [14, 63]. HEMS denotes Helicopter Emergency Medical Services. The "rerouted" event means that a drone is rerouted by the path-planning algorithm due to a conflict with crewed aircraft. The "failure" event in the complex scenario is more challenging to resolve compared to the one in the simple scenario. This is because, in the complex scenario, the drone that fails to reroute has potential conflicts with two crewed aircraft.

Table 5.1: Characteristics of different traffic scenarios in the experiment. Each scenario lasts five minutes.

Scenario	Crewed Aircraft			Drones		Events	
	IFR	VFR	HEMS	Fixed-Wing	Multi-Rotor	Rerouted	Failure
Simple-Normal	3	0	0	2	6	1	0
Simple-Failure	3	0	0	2	6	0	1
Complex-Normal	2	1	1	6	10	4	0
Complex-Failure	2	1	1	6	10	4	1

The experiment took a within-participant design. Each participant was exposed to the 8 measurement scenarios (2 algorithms × 4 traffic scenarios) in a quasi-randomised order to counterbalance presentation order effects. To prevent mixing the algorithms, the four scenarios for one algorithm were treated as a set. Only after completing one set the next set (i.e., another algorithm) can be presented. Therefore, orthogonal Latin Square of order 4 was implemented to achieve a balanced presentation order for the two sets of scenarios. This approach generated four different sequences given the order of the two algorithms. Considering the need to vary the algorithm order as well, the experiment had eight quasi-randomised sequences for the measurement scenarios.

5.4.4. DEPENDENT MEASURES

The dependent measures in the experiment were as follows:

Safety. Although the experiment mainly aimed to explore the usage of transparency in supervising UTM systems, safety was the most important goal that participants needed to achieve. This was measured by the minimum separation between drones and crewed aircraft and the number of loss of separation events.

Efficiency. Similarly, the mission efficiency of drones was the secondary goal for participants. This was measured by the average delay and the number of failed drone missions (i.e., unable to reach its destination). To calculate the average delay, the delay for failed drone missions was defined as the maximum allowable delay instead of being set to infinity. The maximum allowable delay was calculated by the drone endurance minus the minimum flight time (if following a direct route).

Interface usage and preference. The transparency usage was measured by the durations of displayed transparency elements and the number of times detailed transparency information was requested through right-click actions. Participants' preferences regarding the usefulness of the transparency elements for UTM supervision were measured on a five-point Likert scale after the experiment.

Intervention. This was measured by the number of geofence activations and drone commands, as well as the number of times drone control was taken from UTM and delegated back to UTM.

Workload. This was measured by NASA-TLX [237] from six dimensions: mental demand, physical demand, temporal demand, performance, effort, and frustration. The first part of NASA-TLX involves direct ratings on these dimensions, while the second part consists of pairwise comparisons to generate weights for each dimension. The total workload is calculated as the sum of the weighted workload ratings. However, performing the pairwise comparisons after each measurement run would be too time-consuming. Similar to van de Merwe *et al.* [229], the pairwise comparisons were performed only after every four measurement runs (i.e., once per algorithm). The resulting weights were then applied to the four measurement runs to calculate the overall workload for each run.

Interface acceptance. After each measurement run, participants were required to evaluate the interface based on the Modified Cooper-Harper for Unmanned Vehicle Display (MCH-UVD) [238]. In this study, the rating scale was reversed, with 10 representing the best and 1 the worst, similar to the the Controller Acceptance Rating Scale (CARS) [239]. MCH-UVD was selected because it was specifically designed for unmanned vehicle displays, making it well-suited for the UTM case.

5.4.5. CONTROL VARIABLES

The control variables in the experiment were as follows:

Degrees of freedom. Participants could only adjust the drone heading and could not let the drone loiter. They were also not allowed to issue instructions to crewed aircraft.

Drone mission type. All drones in the experiment performed point-to-point delivery missions and were either fixed-wing or multi-rotor types. Unlike Janisch *et al.* [63], the drones were assigned the same priority to keep the focus on transparency usage.

Scenario. The scenarios were identical for Zeta*-SIPP and Informed RRT*. However, to avoid recognition, the scenarios were mirrored, including crewed and drone traffic and wind fields, using the runway centre line as the origin for mirroring.

Algorithm parameter. The parameters of the path-planning algorithms were fixed, with a consistent grid size for Zeta*-SIPP and a fixed termination condition for Informed RRT*. The grid size of Zeta*-SIPP was the same as the geofence grid size, equal to the safe separation between drones and crewed aircraft (assuming 1 km). The parameter settings of Informed RRT* have been mentioned at the end of Section 5.3.2. Although the different algorithms were applied to the same traffic scenarios, the scenarios and algorithm settings were carefully selected to ensure that they produced similar solutions and that the same drones failed to be rerouted.

Initial run setting. At the start of each trial, all transparency elements were deactivated and the flight strip panel was hidden. This ensured that each trial had the same initial conditions for each participant.

5.4.6. HYPOTHESES

It was hypothesised that the conflict detection for the flying route (Element 2) would be the most useful element for UTM supervision (H1). This hypothesis was driven by the previous survey-based user study in Chapter 4, which revealed that operators preferred operational transparency and regarded the conflict detection for the flying route as one of the most useful transparency elements. Based on the study in Chapter 4, it was also hypothesised that failure scenarios would lead to more usage of transparency elements, particularly the explored space and search process, compared to normal scenarios (H2). This was because, when automation failed, operators may seek more algorithmic information to understand what happened. They may also need more support from UTM to complete their supervision tasks.

Moreover, it was also expected that Informed RRT* may be more suitable for UTM supervision compared to Zeta*-SIPP. Specifically, operators would be more willing to access the explored space and search process elements of Informed RRT* for support, which would consequently lead to a reduced workload (H3). Although Informed RRT* employs a random exploration strategy, its search trees may be more intuitive and meaningful for humans. Even if it failed to find a feasible path, it may still provide hints about potential directions to resolve conflicts according to the final exploration results (search trees). For example, as shown in Figure 5.9, Informed RRT* failed to find a path due to its limited number of sampling points. However, the search tree suggested that navigating the drone to the right may have a greater chance to resolve the conflict, as all leftward nodes failed to cross the corridor due to the headwind. This may be particularly useful when operators are not familiar with how to handle this situation. In contrast, while the search tree visualisation of Zeta*-SIPP may appear more organised, it cannot provide clear guidance on which direction a human operator should explore for potential solutions.

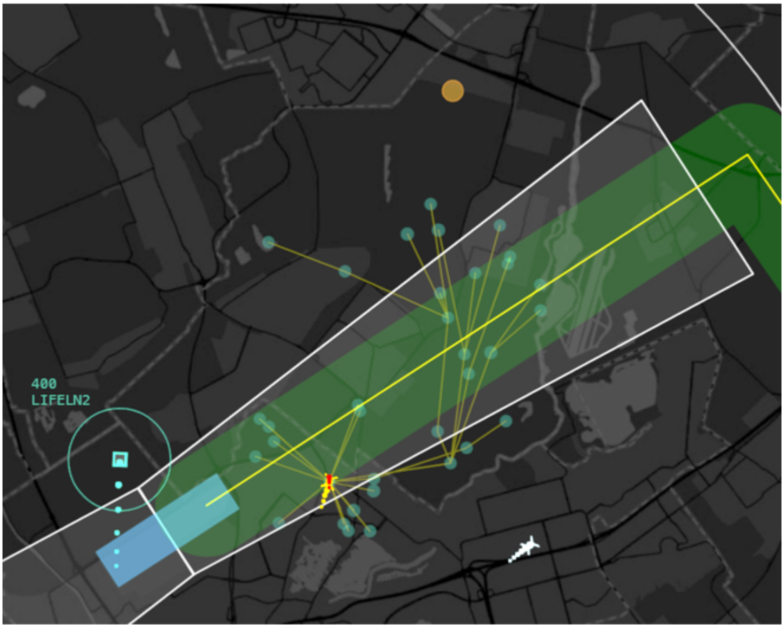


Figure 5.9: A failure event for Informed RRT*

5.5. RESULTS

5.5.1. DATA ANALYSIS AND STATISTICS

Given the relatively small sample size ($N = 16$), conservative non-parametric tests were adopted. To compare Zeta*-SIPP and Informed RRT*, Wilcoxon Signed-Rank tests were conducted for each dependent measure. The matched-pairs rank biserial correlation coefficient rc [200] was then calculated to measure the effect size for Wilcoxon Signed-Rank tests (small ≥ 0.1, medium ≥ 0.3, large ≥ 0.5). To explore the differences among transparency elements or scenarios, Friedman tests were performed, followed by Exact tests [163] with Bonferroni correction for further pairwise comparisons. Kendall's coefficient of concordance w was used to measure the effect size for Friedman tests (small ≥ 0.1, medium ≥ 0.3, large ≥ 0.5). The significance level was set to 0.05. As the effect size reflects the magnitude of the difference between groups [164], it is ideal to have both a statistically significant result ($p < 0.05$) and a large effect size to claim a clear and meaningful difference. A low effect size suggests that the statistical significance should be interpreted with caution.

5

5.5.2. SAFETY AND EFFICIENCY

Figure 5.10 shows the minimum separation and average delay in different scenarios. Only one loss of separation occurred in the simple-failure scenario (the whisker extends beyond the red dashed line) because one participant specifically wanted to ensure the success of drone missions and refrained from using the "Land Immediately" command, even when a drone and a crewed aircraft were particularly close. This decision led to a failure to maintain safe separation. In normal scenarios, human intervention did not consistently improve efficiency compared to the baseline (red solid line). Some participants preferred larger separation to ensure safety. In the simple-failure scenario, 15 out of 16 participants successfully resolved the automation failure in both Zeta*-SIPP and Informed RRT*, leading to a notable improvement in efficiency. In the complex-failure scenario, 11 out of 16 participants resolved the issue in Zeta*-SIPP, while 12 out of 16 did so in Informed RRT*. The efficiency gains were more pronounced in the simple-failure scenario, likely because the smaller number of drones made successful failure resolution more impactful on overall performance. The variations within the minimum separation and average delay are both small in the complex-failure scenario. This is because most participants paid more attention to the drone that failed to reroute, leaving the other drones controlled by UTM without human intervention.

The differences in safety and efficiency between Zeta*-SIPP and Informed RRT* are primarily influenced by their respective parameter settings. For example, increasing the number of samples in Informed RRT* can reduce its delay and bring its minimum separation closer to 1 km. Therefore, the statistical comparisons based on Figure 5.10 may not accurately reflect the human effects. To further analyse the impact of human intervention on safety and efficiency, the increases in the minimum separation and average delay were calculated relative to the baseline results, as depicted in Figure 5.11. The positive direction of the x-axis indicates larger minimum separation, while the positive direction of the y-axis represents increased average delay. Since increasing separation usually comes at the cost of increasing delay, no points appear in the lower-right quadrant.

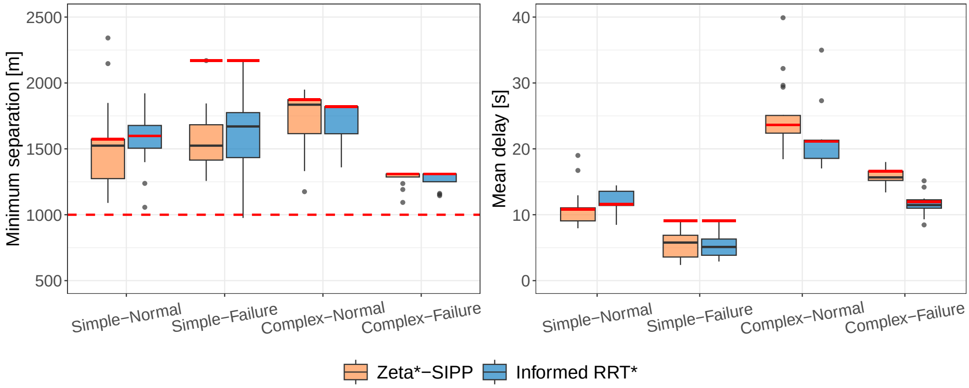


Figure 5.10: Minimum separation and average delay in different scenarios. The red dashed line indicates the assumed safe separation standard. The red solid lines represent the baseline results when no human intervention is involved.

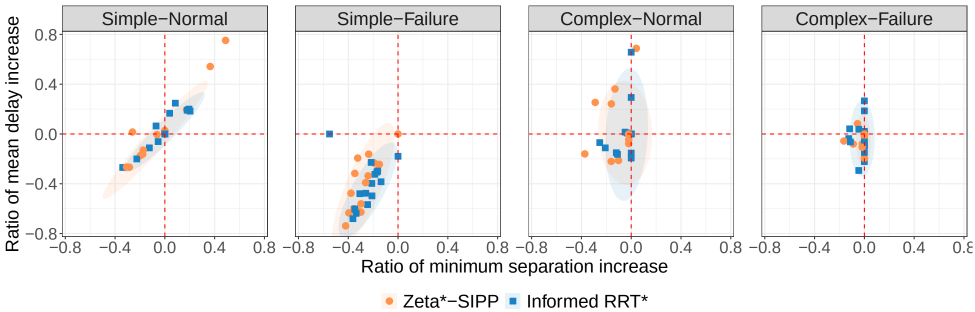


Figure 5.11: The impact of human intervention in terms of minimum separation and average delay, featuring 95% confidence ellipses.

No significant difference was found between Zeta*-SIPP and Informed RRT* across all scenarios regarding the ratio of mean delay increase and the ratio of minimum separation increase. It is evident that there is an approximate linear relationship between these two metrics in the simple scenarios. This is because, in these scenarios, only one drone was designed to be affected (see Table 5.1), making the impact of human intervention more pronounced compared to the complex scenarios. In the simple-failure scenario, all points cluster in the lower-left quadrant because almost all participants successfully resolved the failure event. In the experiment, two distinct strategies were identified. Some participants attempted to reduce the separation to minimise delays (lower-left quadrant), while others tended to add more separation buffer to enhance safety (upper-right).

5.5.3. USAGE AND PREFERENCE

During the experiment, participants were provided with eight buttons to access different types of transparency information. Since the computation of the first six elements (buttons) was independent of the path-planning algorithm, their usage primarily depended on the participants' preferences and the specific situations they encountered. Therefore, to compare the usage of the eight transparency elements, their average display durations were first calculated for Zeta*-SIPP and Informed RRT*, as shown in Figure 5.12. In addition to the geofence grids and wind field, the other six transparency elements are generated based on drone routes. This means that accessing these elements requires not only activating the buttons but also selecting a drone to display its route. Thus, the display durations of transparency elements in Figure 5.12 have been corrected by the display duration of drone routes.

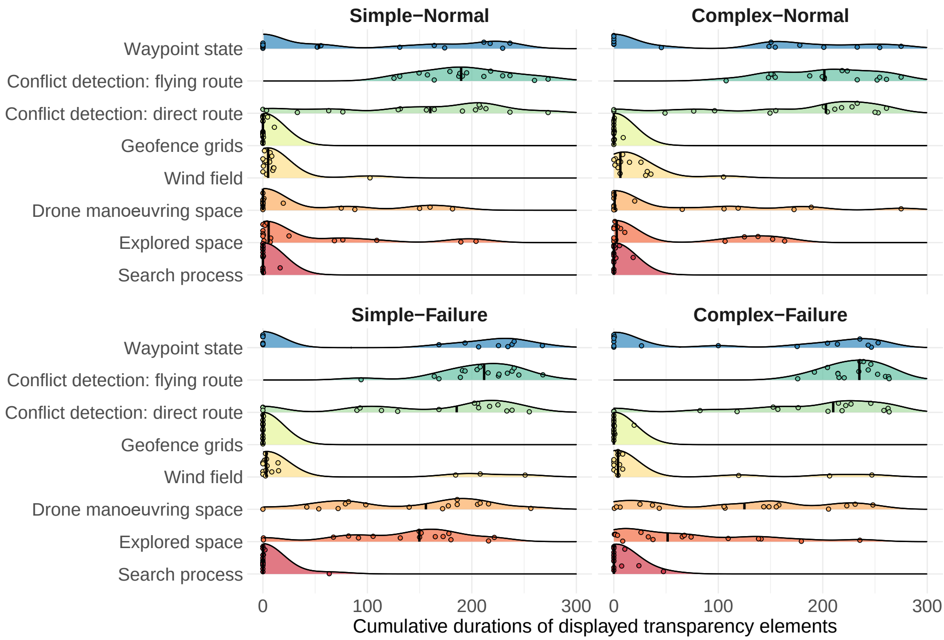


Figure 5.12: Ridgeline plots of transparency element display durations, with black vertical lines marking the median values. Each dot represents a participant.

From Figure 5.12, it is clearly seen that the conflict detection for the flying route and the direct route was the most frequently used information in the experiments. Many participants activated these two elements immediately at the start of the scenarios, drawing on their previous experience from the training round. In contrast, the usage of the waypoint state was highly polarised. Some participants activated the waypoint state button simultaneously with the conflict detection buttons, while the others almost never used it.

The geofence grid was the least used transparency information. This is because participants rarely used geofences for conflict resolution and instead favoured active control over specific individual drones. Also, the size and shape of the geofence grids remained consistent across all scenarios. Participants might have already been familiar with this information during training and chose to disable it to maintain a clean interface. The use of the wind field element was also limited. Participants tended to glance at the wind field, particularly when considering intervention in drone routes, and then immediately disabled it. This suggests that the wind field could be valuable for UTM supervision, but the abundance of coloured arrows and animated particles may contribute to visual clutter.

The drone manoeuvring space and explored space were utilised more frequently in the failure scenarios. Participants primarily used them as supporting information to diagnose the failure events. The drone manoeuvring space represents the additional battery capacity available for drone rerouting, while the explored space reflects the efforts made by the algorithm to resolve conflicts. The display durations of these two elements were shorter in the complex-failure scenario compared to the simple-failure scenario. This is likely due to the higher information load in the complex-failure scenario, where participants tended to disable these elements after addressing the problem. The search process was rarely used, with its slider accessed in only 6.25% of the experimental runs (16 participants × 8 measurement runs).

As the use of geofence grids, wind field, and search process is limited, Friedman tests were conducted only on the remaining five transparency elements. The results revealed significant differences among them in terms of the display durations ($\chi^2(4) = 29.123, p < 0.001, w = 0.455$). Pairwise comparisons with Bonferroni correction further disclosed that the conflict detection for the flying route (Element 2) was significantly different from the waypoint state ($D = 34, p < 0.001$), drone manoeuvring space ($D = 39, p < 0.001$), and explored space ($D = 41, p < 0.001$). The difference between conflict detection for the flying route and the direct route warrants further comparison.

As for the traffic scenarios, Friedman tests found significant differences among them in the conflict detection for the flying route ($\chi^2(3) = 13.189, p = 0.004, w = 0.275$), drone manoeuvring space ($\chi^2(3) = 32.396, p < 0.001, w = 0.675$) and explored space ($\chi^2(3) = 19.777, p < 0.001, w = 0.412$). Pairwise comparisons with Bonferroni correction were then conducted, with the results presented in Table 5.2. No significant difference between the simple-normal and complex-normal scenarios was found, suggesting that traffic complexity may not be the primary factor influencing the usage of transparency. The differences in the transparency usage mainly stemmed from the automation failure events. Participants generally required more information, particularly the drone manoeuvring space and algorithm's explored space, to diagnose the failures. In fact, these transparency elements not only helped participants understand the rationale behind the failures but also improved their situation awareness and supported more informed decision-making. For example, in the experiment, when automation failed to resolve a conflict, some participants requested the drone manoeuvring space in order to consider the battery power limitations for manually navigating the drone.

To further compare the differences between Zeta*-SIPP and Informed RRT*, Figure 5.13 presents the display durations of the explored space for these two algorithms. In the normal scenarios, usage of the explored space between Zeta*-SIPP and Informed RRT* is similar, with median values for both being close to zero. In the failure scenarios, participants relied more frequently on the space explored by Informed RRT*. This is because it provided participants with a clue that was more helpful in solving the problem. This benefit was explicitly acknowledged in the comments of the ten participants who preferred Informed RRT* over Zeta*-SIPP. For example, one participant stated that Informed RRT* "*gave me more intuition on where to look for feasible solutions*", and another one

Table 5.2: Pairwise comparisons of the display durations among scenarios (with Bonferroni correction).

Scenario	Detection: flying route			Drone manoeuvring space			Explored space		
	SN	SF	CN	SN	SF	CN	SN	SF	CN
Simple-Normal (SN)									
Simple-Failure (SF)	1.000			***			0.002**		
Complex-Normal (CN)	1.000	1.000		1.000	***		1.000	0.006**	
Complex-Failure (CF)	0.006**	0.136	0.016*	***	1.000	0.006**	0.025*	1.000	0.040*

Note: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

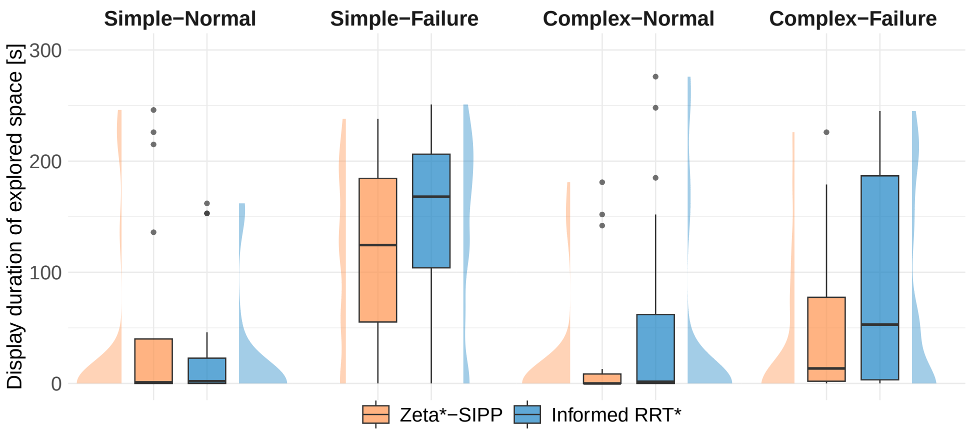


Figure 5.13: Differences in the use of the Explored Space element between Zeta*-SIPP and Informed RRT*.

noted that "*especially in uncertain cases it [Informed RRT*] helps to see which region contains most plausible solutions*". Nonetheless, no significant difference was found between these two algorithms in all scenarios. Participants also used the explored space of Zeta*-SIPP to gather more information in the failure scenarios.

In addition to viewing the transparency elements, participants also accessed detailed information regarding waypoints, CPA, wind vectors, and cost values by right-clicking on the corresponding elements. Almost no one accessed the wind vectors or cost values. These two types of information may have been overly detailed for the UTM supervision task in this particular experiment. Figure 5.14 presents the number of waypoint and CPA access events. While some participants activated the waypoint state element (see Figure 5.12), they rarely accessed the detailed information. It appears that they merely wanted green points to make the waypoints more visually salient. The difference between conflict detection for the flying route and the direct route is more pronounced in terms of the number of information requests. Wilcoxon Signed-Rank tests confirmed the significant difference between these two elements ($V = 136, p < 0.001, rc = 1$). This illustrates that participants relied mainly on the conflict detection for the flying route (Element 2) to monitor the situation. The conflict detection for the direct route was primarily utilised when participants intended to direct a drone straight to its target. The number of accesses

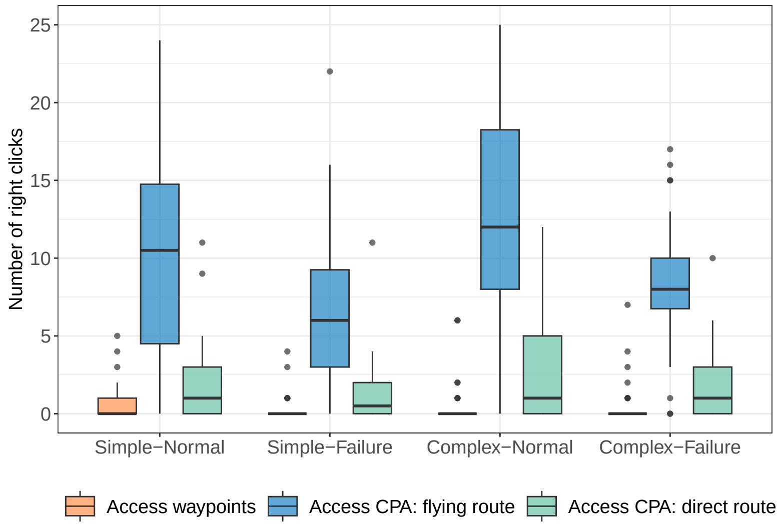


Figure 5.14: Number of information requests concerning the waypoint and CPA details.

to the CPA for the flying route was lower in the failure scenarios because participants put more effort into solving the problem and may not have enough time to access and digest the detailed information.

Figure 5.15 shows the subjective ratings from participants for the eight transparency elements, which align with their actual usage depicted in Figure 5.12. The geofence grids and search process were considered relatively irrelevant for supervising the UTM system in this experiment. They may be more valuable for training, helping operators become familiar with the geofence grids and understand how the algorithm works. Regarding the waypoint state element, the detailed information seems to have limited usefulness, as suggested by Figures 5.14-5.15. However, it could be more useful if it included more details, such as altitude changes, which are not directly observable from a 2D interface. In this experiment, drone altitude was not considered, making the waypoint state element less useful. However, in urban environments, drone altitude may require greater attention to avoid collisions with high-rise buildings. Operators may need to inspect the waypoints to ensure that the planned path remains above the terrain. Therefore, the information elements that were not accessed or requested in this experiment should not be considered completely useless. They may just be less important for successfully accomplishing the task in this particular context.

5.5.4. INTERVENTION

Participants were provided with different means to intervene in drone traffic, including geofence activations. However, geofences were utilised in only 7.81% of the experiment runs. In most cases, participants favoured active control over specific individual drones. This is because geofences could impact multiple drones simultaneously and thus may not be very efficient. Additionally, the number of drones in the experiment was relatively

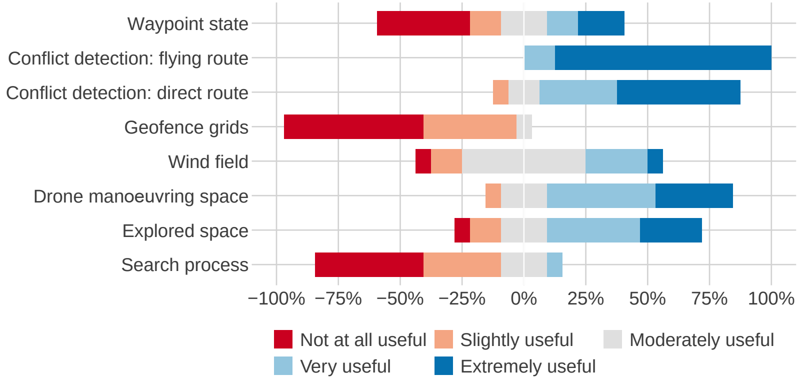


Figure 5.15: Likert scale ratings for the eight transparency elements regarding their usefulness in supervising the UTM system.

low, making the active control still manageable, particularly with the support of a high level of automation (path-planning algorithm). As shown in Figure 5.5, operators can issue commands to drones only after assuming control. Therefore, the number of takeovers reflects the extent of intervention and the effort exerted by participants, while the number of re-delegations to UTM indicates their trust in the system and willingness to return control, as depicted in Figure 5.16. No significant difference was found between Zeta*-SIPP and Informed RRT*. In the normal scenarios, around 30% participants did not take any control actions. They monitored the traffic and concluded that safety could be ensured without any intervention. In the failure scenarios, approximately 40% participants intervened only once with the drone that failed to reroute.

To further compare the differences among the four scenarios, the average number

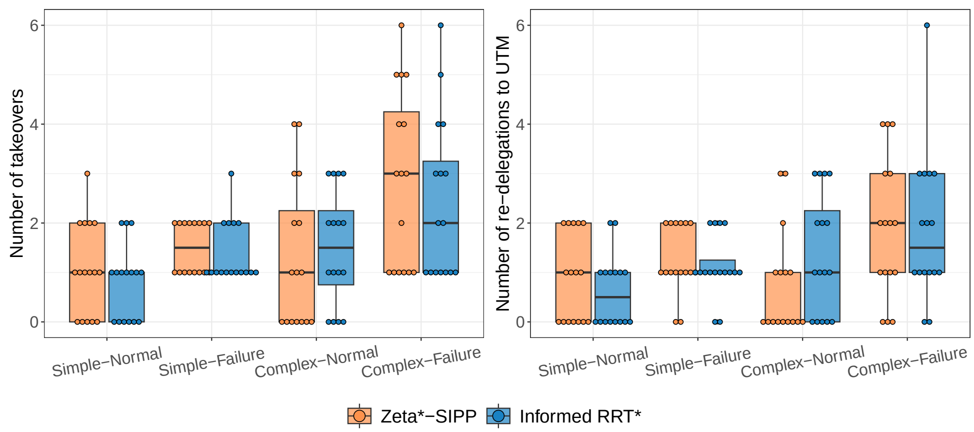


Figure 5.16: Number of takeovers and re-delegations to UTM.

of control take overs was calculated for Zeta*-SIPP and Informed RRT*. Friedman tests then revealed significant differences between them ($\chi^2(3) = 21.448, p < 0.001, w = 0.447$). Pairwise comparisons with Bonferroni correction further disclosed that the complex-failure scenario was significantly different from the simple-normal ($D = -32.5, p < 0.001$) and complex-normal ($D = -22, p = 0.016$) scenarios. This is because, in the complex-failure scenario, some participants repeatedly took over control and delegated it back to UTM, falling into a reactive loop without a clear solution to the problem. They wanted to reroute drones by actively triggering the path-planning algorithm, but the limitations of the algorithm often prevented it from functioning as intended, especially when it had already failed in the first place. When UTM (re)routing fails, the system should probably monitor for changes in state continuously to determine whether UTM can resume control. Once a viable solution is found, a prompt can be displayed to inform operators and allow them to choose whether to accept it. This way, operators will understand that if no prompt appears, the UTM routing service has not identified a solution – thereby avoiding inefficient reactive loops.

5.5.5. WORKLOAD

Figure 5.17 shows the overall workload ratings in different scenarios measured by NASA-TLX. Generally, Informed RRT* results in a lower workload than Zeta*-SIPP, with the difference being particularly pronounced in the complex-failure scenario. Wilcoxon Signed-Rank tests confirmed the significant difference in this scenario ($V = 96, p = 0.044, rc = 0.6$). This is likely because the explored space of Informed RRT* could offer more support for participants in addressing failure events. As the drone's battery power decreases, its manoeuvring space (i.e., search space) narrows. Given the fixed number of sampling points, the likelihood of finding a feasible path with Informed RRT* actually increases. To further compare the differences among the four scenarios, the average workload was calculated for Zeta*-SIPP and Informed RRT*. Friedman tests revealed significant differences between them ($\chi^2(3) = 33.075, p < 0.001, w = 0.689$). Pairwise comparisons with Bonferroni correction further disclosed that the simple-normal and complex-failure scenarios differed significantly from all the other scenarios. This illustrates that the workload for supervising the UTM system is influenced by the scenario complexity.

Figure 5.18 presents the distribution of the workload measured by NASA-TLX across six dimensions. For easier observation, the ratings are normalised to $[0.1, 1.0]$. Unlike the original package [240], the max-min normalisation is applied to the entire data matrix instead of individually to each dimension, as all dimensions share the same scale. As the complexity of the scenario increases, nearly every dimension of the workload increases as well. However, the physical demand drops significantly in the complex-failure scenario for both Zeta*-SIPP and Informed RRT*. This is likely because participants focused more on the drone that failed to reroute, reducing their attention to and interventions with other drones. This trend can also be observed in Figures 5.10 and 5.14. For Zeta*-SIPP, the temporal demand was consistently higher than that of Informed RRT* across all scenarios. This can be attributed to the fact that the explored space of Zeta*-SIPP is more ambiguous [214], requiring participants to allocate more attention to interpreting the portrayed information. For Informed RRT*, the frustration was generally higher, probably because the uncertainty introduced by its random sampling strategy caused

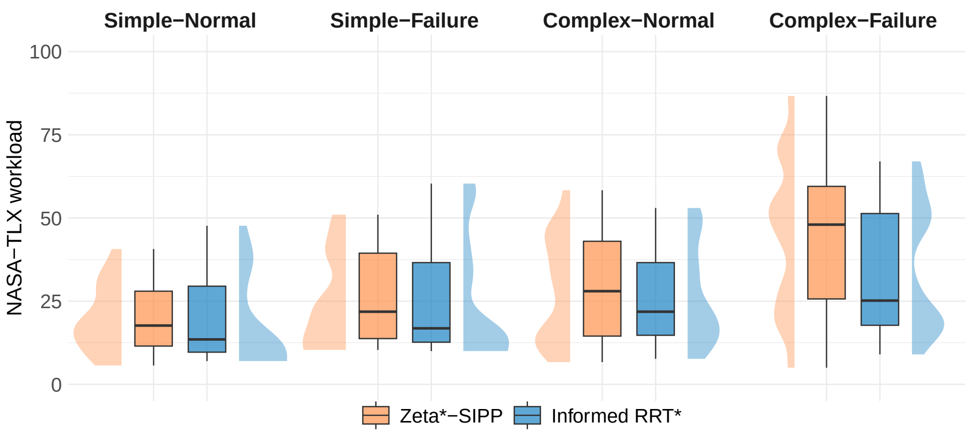


Figure 5.17: Overall workload ratings measured by NASA-TLX in different scenarios.

participants to feel insecure.

5.5.6. INTERFACE ACCEPTANCE

Table 5.3 presents the participants' subjective ratings of interface acceptance as measured by MCH-UVD. Please recall that a rating was given after each trial, and thus there are totally 128 results. In most cases (90.6%), participants considered the display acceptable for supervising UTM operations. Three participants felt that the interface required improvement (ratings of 5 and 6) in the complex-failure scenario. This was caused by Zeta*-SIPP, and the participants believed that Zeta*-SIPP and its associated transparency information (e.g., explored space) did not provide adequate support for task completion.

Table 5.3: Number of observations in the MCH-UVD categories.

Scenario	Display is acceptable		Deficiencies warrant improvement		Deficiencies require improvement				Mandatory redesign	
	10	9	8	7	6	5	4	3	2	1
Simple-Normal	17	15	0	0	0	0	0	0	0	0
Simple-Failure	17	13	2	0	0	0	0	0	0	0
Complex-Normal	15	16	1	0	0	0	0	0	0	0
Complex-Failure	8	15	4	2	2	1	0	0	0	0

In the complex-failure scenario, some participants found it difficult to address the failure event based on the available information. The interface should be improved to better support operators in handling automation failures. For example, instead of repeatedly issuing commands and observing the outcomes, a preview feature could be in-

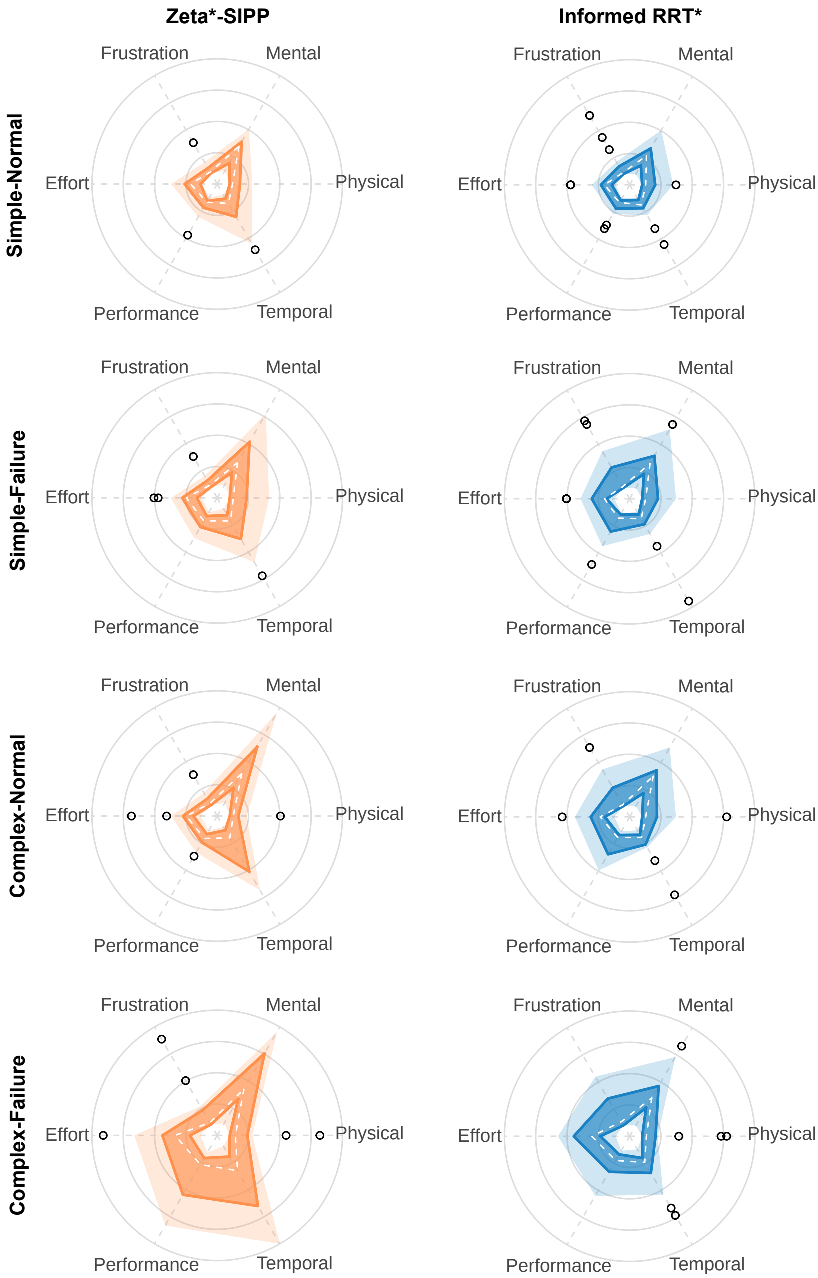


Figure 5.18: Radar-boxplots [240] depicting the distribution of workload across six dimensions, with weights for each dimension integrated into the ratings. The inner darker region represents the 25th to 75th percentiles of each dimension, while the outer lighter region indicates the total range, excluding outliers. The white dashed line denotes the median value.

troduced that allows operators to quickly review potential resolution options before finalising their decisions. As one participant noted, "*It would be nice to have a preview for what the UTM would plan, because sometimes I switched back to the UTM but then regretted it.*" A drop-down menu with predefined headings (e.g., with 5° increments) can also be integrated into the command line. When operators hover over a heading option, a preview of the outcomes (e.g., CPA) can be displayed on the main screen. However, the downside of such a preview feature is that it may cause delays in operator responses. More experiments are required to explore its impact.

Another approach is to allow operators to directly manipulate drone routes on the map, accompanied by a preview that indicates whether a solution is viable or problematic. As one participant suggested, "*It would be nice if, instead of setting a heading, you could click to add an intermediate waypoint.*" This approach follows the principles of direct manipulation [241] and is similar to existing ATM decision support tools [109, 187, 195, 242]. Additionally, the interface would be more flexible if it allowed operators to adjust the parameters of the path-planning algorithm. A participant remarked, "*Zeta*-SIPP would be better if the grid size were smaller in my opinion.*" If feasible solutions exist in the search space, an automation failure in path planning typically indicates that the algorithm is overly constrained by its own parameters, such as the grid size in Zeta*-SIPP and the number of sampling points in Informed RRT*. In this case, human operators could either manually examine areas beyond those explored by the algorithm or enhance the algorithm's capabilities to explore a broader space.

In summary, these interface deficiencies primarily arise from insufficient decision support for human operators. This is because the interface developed in this study was mainly designed for supervision, particularly through various transparency elements. Further research is needed to enhance support for human intervention.

5.6. DISCUSSION

5.6.1. THE UTILITY OF TRANSPARENCY IN UTM SUPERVISION

In this research, eight transparency elements were designed for UTM based on the unified transparency taxonomy (see Figure 5.1). A human-in-the-loop experiment was conducted to explore the usage of these elements in UTM supervision. The results indicate that the conflict detection for the flying route was found to be the most important and useful transparency element (H1). This finding aligns with the previous survey study in Chapter 4 and the design choice implemented in the TAPAS project [55]. Unlike traditional ATM, trajectory prediction and conflict detection in UTM are much more challenging for operators because of the substantial speed differences between drones and crewed aircraft. Most participants heavily relied on, and frequently used, the conflict detection element during the experiment. This phenomenon has also been confirmed in another UTM experiment [63].

The experiment results also suggest that operators require more information when automation fails, particularly the drone manoeuvring space and the algorithm's explored space (H2). Contrary to the expectations, participants rarely accessed the algorithm's search process, even in the failure scenarios. During the experiment, it was observed that some participants felt confused when UTM failed to reroute a drone. However, when they noticed the appearance of a new crewed aircraft whose route conflicted with the drone's route, they quickly understood what happened and why UTM failed. Although the corresponding event messages appeared in the bottom-left corner of the display, participants often overlooked them, focusing instead on the map. It seems that, instead of detailing the search process, highlighting changes in the environment or constraints directly on the map may be more useful for explaining automation failures in operational contexts [77]. The presentation of the search process may be more suitable for training and post-hoc analysis, helping operators become familiar with their algorithmic co-workers.

It was found that participants mainly used transparency information to enhance their situation awareness, such as actively scanning the CPA-related information to detect potential conflicts and activating the wind field and drone manoeuvring space to identify domain constraints. Even though the transparency elements were designed primarily to provide insights into UTM, participants also utilised them as a decision support tool. For example, most participants disabled the algorithm's explored space in the normal scenarios. However, in the failure scenarios requiring operator intervention, participants examined the explored space (the algorithm's efforts to find solutions) to gather hints about which direction was most likely to resolve conflicts and enable the drone to complete its mission, especially when using Informed RRT*.

Additionally, transparency helped participants gain a clear understanding of the algorithms and their limitations. Regarding Zeta*-SIPP, one participant observed, "*It starts with the direct route and then searches for small deviations from that.*" Another participant recognised how to revise the algorithm to improve its performance: "*Zeta*-SIPP would be better if the grid size were smaller in my opinion.*" For Informed RRT*, a participant shared, "*It gave me more intuition on where to look for feasible solutions*". Another noted, "*It cannot ensure that the solution found is optimal.*" This benefit of transparency on human understanding aligns with findings from the previous study in Chapter 3.

5.6.2. THE IMPACT OF ALGORITHM TYPE ON UTM SUPERVISION

Graph- and sampling-based algorithms are two common approaches for path planning. Both first discretise the continuous search space to reduce the complexity of the problem and then perform a search within the discretised space to build a search tree. The branches of the tree represent passable connections between nodes, and the path from the start (root) to each node can be traced through these branches. Once the target is added to the search tree, it means that a feasible path from the start to the target has been found. For optimal path-planning algorithms, such as those based on A*, the feasible path found by the tree typically represents the optimal path within the discretised space. An advantage of graph- and sampling-based path-planning algorithms is that their internal processes can be easily portrayed to achieve algorithmic transparency. This benefit could make it easier for operators to understand how the algorithms work and accept them as collaborators in tactical UTM operations. This is also why Zeta*-SIPP and Informed RRT*, two advanced graph- and sampling-based algorithms, were chosen for UTM rerouting in this research.

The experiment results suggest that Informed RRT* might be better suited than Zeta*-SIPP for operators supervising the UTM system (H3). Although no significant difference was found in the use of the explored space element, participants tended to activate this element more frequently when using Informed RRT*. Additionally, the workload associated with Informed RRT* is generally lower compared to Zeta*-SIPP, with a significant difference observed in the complex-failure scenario. The post-hoc questionnaire shows that 10 out of 16 participants preferred Informed RRT*, appreciating its ability to provide potential directions for conflict resolution and rerouting.

However, this cannot conclusively recommend that the UTM system should adopt a sampling-based path-planning algorithm. Different operators may have different preferences. The post-hoc questionnaire also reveals that 6 participants favoured Zeta*-SIPP, disliking the randomness of Informed RRT* and valuing the optimality of Zeta*-SIPP. One possible solution is to leverage the strengths of both algorithms, allowing operators to choose their preferred option during operations. Zeta*-SIPP can be used as the default option. Should it fail to reroute, operators can switch to Informed RRT* to explore potential directions for resolving conflicts. Certainly, this method applies only to centralised UTM systems, where one algorithm controls all drones instead of each drone having a different algorithm. The differences between centralised and distributed UTM systems will be discussed in Section 5.6.3.

Additionally, operators can be allowed to specify a direction for (Informed) RRT* exploration, enabling its sampling to follow a Gaussian distribution centred on that direction rather than being uniformly random within its sampling space. This could make the operators' intent transparent to UTM as well, supporting bi-directional transparency [134]. This approach allows the algorithm to explore the search space in alignment with the operators' preferences and may also be helpful in scenarios involving automation failures. For example, instead of relying on manual trial and error, operators can assign a potential direction for resolving conflicts to the algorithm and let it explore automatically. However, this customisation could also slow down the algorithm's convergence to the optimal solution, potentially affecting its efficiency and increasing flight delays.

Overall, the choice of algorithms and whether to tune them to cater to human preferences depend on the system's goals. For a human-centred automated system like ATM and UTM, aligning algorithmic behaviour with users' needs, expectations, and decision-making processes may reduce their doubts, lower their demands for transparency, and increase their acceptance and trust in the automated system [61, 113, 201].

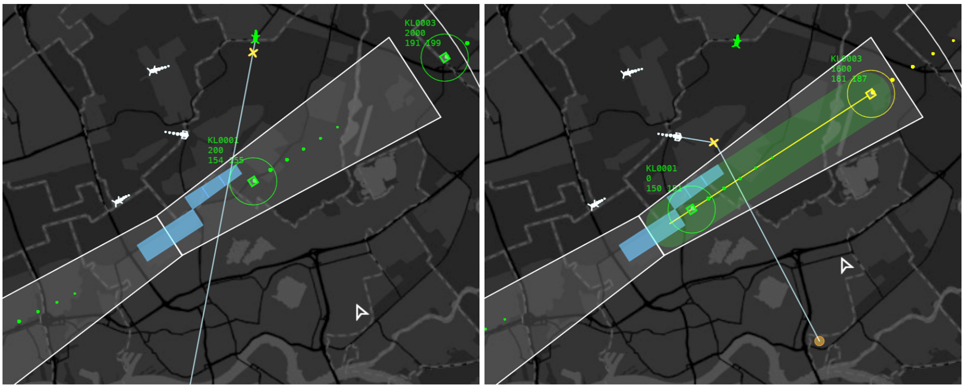
5.6.3. THE CONCEPT OF OPERATIONS FOR UTM IN CTR

During the experiment, participants were allowed to intervene in the UTM system by using geofences or assuming direct control over drones. Actually, the extremes of these two methods correspond to two operational concepts: full segregation and full integration [15]. The full segregation is airspace-centric, indicating that the responsibility over the airspace should be clarified between ATM and UTM. The ATM airspace is regarded as no-fly zones (geofences) for UTM operations. The full integration resembles the current ATM system, which resolves conflicts by directly modifying aircraft routes (e.g., issue a heading command). However, since UTM is fully automated, operator intervention involves the allocation of responsibility for drone flights, which is a flight-centric approach [235].

The experiment results show that participants rarely used geofences and preferred direct control of drones. This finding aligns with the observations in the previous UTM experiments [14, 15, 63], where operators expressed their wish to actively influence UTM routing to resolve conflicts. However, this does not mean that geofencing is useless. If the traffic load would be very high, operators may not have sufficient time to directly control drones. In this experiment, the number of drones was relatively low (not hundreds), and most drones were safely managed by UTM since the path-planning algorithm took dynamic obstacles into account as well. Participants only needed to intervene with a few drones they found problematic.

The algorithm requires predicting the trajectories of crewed aircraft. If the prediction is not accurate, the routes found by the algorithm may be unsafe. Should this uncertainty affect multiple drones, operators might need to activate geofences to safeguard crewed aircraft from drone interference. Technically, a dynamic obstacle indicates that a certain space will be occupied over a certain period. Geofencing actually adds a time buffer to this occupation, which is particularly helpful when the predictions of dynamic obstacles are unreliable. In real-world operations, HEMS and VFR flights may be more uncertain than those simulated in the experiment, and hazardous weather patterns, such the movement of strong wind fields, are difficult to predict as well. In this case, geofencing could serve as a tool to ensure that drones are safely separated from other objects.

During the experiment, it was also found that one participant combined geofencing with manual control, as shown in Figure 5.19. They activated geofences to increase the separation buffer for a drone with KL0003, while simultaneously instructing another affected drone (green drone) to ignore the geofences and proceed directly to its target. Overall, the effectiveness of the two control methods, Dynamic Airspace Reconfiguration (DAR) and issuing direct commands to drones, across different scenarios may warrant further research.



(a) Green drone passing through the activated geofences

(b) Blue drone affected by the activated geofences

Figure 5.19: Geofencing combined with manual control in the simple-normal scenario: screenshots from a participant's experiment. Both KL0001 and KL0003 are preparing to land. The route and corresponding green corridor of KL0003 are shown in (b).

In addition to geofencing, which prevents drones from entering a designated area, geocaging has been introduced to ensure that drones remain within a specified region [11]. For example, if a certain airspace is rarely accessed by crewed aircraft, it can be allocated for drone operations through geocaging, thereby alleviating concerns about interactions between drones and crewed aircraft. This approach is particularly appropriate for the initial implementations of UTM, when drone traffic is still relatively low. However, as drone operations continue to expand, it becomes inevitable that they will need to operate within airspace occupied by crewed aircraft during activities such as railway inspections and medical deliveries. In some cases, emergency helicopters may also need to enter areas assigned for drone operations. Dynamic geofencing, as demonstrated in this research, is therefore necessary to ensure safe separation between drones and crewed aircraft, especially when the number of drones is large. Future airspace management for drone operations may benefit from combining both geofencing and geocaging.

Additionally, this study adopts a centralised UTM operational concept in controlled airspace (see Figure 5.4), where the responsibility for safe separation between drones and crewed aircraft is managed by a centralised (human or machine) agent, aligning with the U-space framework [9]. However, a distributed UTM system is also possible [8], where drone operators or the autonomous drones themselves are responsible for their flight safety. In this case, the UTM services only provide information assistance (e.g., the locations and intents of other aircraft) and decision support (e.g., conflict resolution advisories) without issuing mandatory commands. Despite the differences between the centralised and distributed UTM systems, the interface developed in this research can also be adapted for supporting drone operators to supervise their own operations [178]. Instead of monitoring the entire airspace, they only need to oversee their individual flights, with the proposed transparency elements offering insights into the automation.

5.6.4. Limitations and future research

To explore the usage of transparency in supervising UTM systems, an interface was developed that served as a test bed for investigating the efficacy of various visual elements. During the experiment, several deficiencies of the interface have been identified. For example, in the interface, clicking on an empty point on the map will deselect all aircraft. Participants often accidentally did it when trying to select an aircraft, inadvertently removing the route information of other aircraft of interest. To access the CPA information, users have to left-click on a drone to display its route and then right-click on the CPA symbol to view the details. Some participants mentioned that this process required too many clicks and suggested that it would be better to display the CPA information directly. Additionally, participants often had to try to input multiple headings to find a satisfactory one using the command line. To mitigate this issue, the interface could allow users to directly manipulate drone routes on the map. Alternatively, it could integrate a dropdown menu with multiple candidate headings into the command line, accompanied by a preview showing the estimated outcomes for each option.

This study focused on visual transparency for automation failures rather than textual transparency, although it did incorporate event messages into the interface (see Figure 5.3). Textual transparency could generally offer more concise explanations for failure causes – for example, "*UTM failed to reroute due to too large grid size.*" However, failures often result from multiple contributing factors. It can also be, "*UTM failed to reroute due to insufficient drone endurance, strong headwinds or incoming crewed aircraft.*" Sometimes, it is difficult to extract clear and unambiguous textual explanations. Moreover, visual transparency is built on the map and is directly tied to drone routes, making it more immediately noticeable. The visual representation of the algorithm's explored space can also provide intuitive cues for human intervention, such as identifying which headings or regions could be conflict-free.

In general, the developed interface was primarily intended to support UTM supervision and monitoring. However, the experiment results indicated that effective support for intervention is also important. When automation fails, operators have to assume control and intervene manually. However, the occurrence of such failures usually signals that little margin remains to resolve the issues. Operators may be overwhelmed by the sudden surge in stress and workload. In these high-pressure situations, a well-designed decision-support tool may be necessary and perhaps even more so than in non-automated settings [169]. In fact, decision-support tools share some similarities with transparency, as both provide relevant information to users. However, decision-support tools primarily focus on assisting users in making *informed* decisions, while transparency emphasises explaining how automation reaches its decisions. Research on explainable motion planning [123] also emphasises: explanations should be actionable, offering guidance on what changes are needed to achieve the desired output. These findings collectively underscore that supporting intervention (e.g., decision-support tools) is as critical as supporting supervision (e.g., transparency).

Future research could consider other types of path-planning algorithms and/or explore potential combinations of graph- and sampling-based algorithms to better satisfy operators' preferences. One could also explore how operators interact with the algorithm when given the ability to adjust its parameters. By intervening in the algorithm, operators can effectively communicate their needs to the system, ultimately fostering better human–machine collaboration. However, incorporating human preferences may reduce the algorithm's efficiency. For example, operators may add a buffer to the minimum separation constraint to encourage the algorithm to find more robust and safer solutions. This decision may reduce their workload and pressure but could also lead to increased delays for aircraft. Therefore, a balance between accommodating human preferences and optimising system performance needs to be found for human-centred UTM systems.

5.7. CONCLUSION

This chapter focuses on UTM in CTR around airports, addressing challenges arising from the collaborative operations between ATM and UTM in low-altitude airspace. To support operators in supervising UTM, a transparent UTM interface was developed that integrates eight information elements derived from a unified transparency taxonomy. A human-in-the-loop experiment was then conducted to explore the usage of these elements in UTM supervision. The results indicate that the conflict detection for the flying route is the most useful element among all scenarios. When UTM fails to reroute a drone, operators tend to require more information, especially the drone manoeuvring space and the algorithm's explored space, to help them manually address the issue. The number of drones may not be the primary factor influencing transparency usage, although it could affect the operators' workload.

The experiment also investigated the impact of algorithm type on UTM supervision, revealing that the sampling-based algorithm Informed RRT* might be more suitable for supervising UTM compared to the graph-based algorithm Zeta*-SIPP. This is because the explored space of Informed RRT* could offer more insights to assist operators in handling UTM rerouting failures. However, the randomness of the sampling-based algorithm may be an obstacle for some operational users to accept it. Integrating both graph- and sampling-based algorithms into UTM might provide more flexibility for supporting UTM supervision. This research serves as a practical example of human-machine interaction with advanced path-planning algorithms in tactical operations, offering a reference for applying transparency in dynamic scenarios.

6

DISCUSSION AND CONCLUSIONS

The previous chapters constitute the main part of this dissertation. They followed a bottom-up approach that began with transparent path planning (Chapters 2 and 3) and gradually extended to its application in UTM (Chapters 4 and 5). This chapter, as a final summary, first revisits the research problem and summarises the main findings by answering the four research questions investigated in this dissertation. It then recommends potential directions for extending this research by relaxing some of the underlying assumptions. Finally, it concludes by distilling the main takeaways and the most significant insights.

6.1. RETROSPECTIVE

To accommodate the anticipated surge in drone operations, with varying types and missions, Uncrewed Air Traffic Management (UTM) systems are being developed worldwide, including American UTM [8] and European U-space [9]. Despite differences among these initiatives, there is a broad consensus that UTM operations should be highly automated. However, it cannot be guaranteed that automated UTM will always function correctly. To enhance the overall safety and reliability, UTM should be designed to enable human operators to supervise its operations [129, 169]. Once UTM fails or behaves unexpectedly, operators should be able to understand the underlying causes and intervene in the UTM decisions when necessary.

Research suggests that some form of "seeing-into" transparency may be necessary to support human supervision of automation [18–20]. However, there is no unified method for transparency design in algorithmic systems, and it remains unclear what algorithmic transparency entails in UTM contexts. Therefore, the main research problem this dissertation addressed is:

Research Problem

How can algorithmic transparency be achieved in UTM to support tactical UTM operations within controlled airspace, ensuring that human operators can effectively understand and supervise automated UTM decision-making?

6

As illustrated in Chapter 1, (conflict-free) path planning is central to UTM since the goal of UTM is to safely navigate drones to their destinations. This dissertation therefore focuses on path-planning algorithms. Based on this, four key points can be derived from the research problem, leading to the formulation of the four research questions:

1) how to achieve algorithmic transparency in path planning (RQ1),
2) how algorithmic transparency improves human understanding (RQ2),
3) how to achieve algorithmic transparency to support UTM supervision (RQ3), and
4) how algorithmic transparency supports UTM supervision (RQ4).

Considering that path-planning algorithms typically follow fixed procedures largely independent of context, a bottom-up approach was adopted, starting with transparent path planning (RQ1 and RQ2) and gradually extending it to the UTM context (RQ3 and RQ4). Figure 6.1 depicts the design cycle of this research.

The first iteration focused on transparent path planning, and aimed to visually reveal the internal processes of path-planning algorithms (Chapter 2). A user study was performed to evaluate its impact on human understanding, demonstrating its effectiveness. (Chapter 3). In this iterative process, no specific context was involved, and the resulting transparency was referred to as *engineering transparency*.

In the second iteration (Chapter 4), the scope was expanded to the UTM context, introducing the concept of *operational transparency* to provide information that supports operator situation awareness. A unified transparency taxonomy was thus proposed by integrating operational and engineering transparency, as shown in Figure 6.2. In this taxonomy, *domain transparency* was defined as a bridge, referring to the disclosure of the physical and intentional constraints of work domains. Based on the taxonomy, various

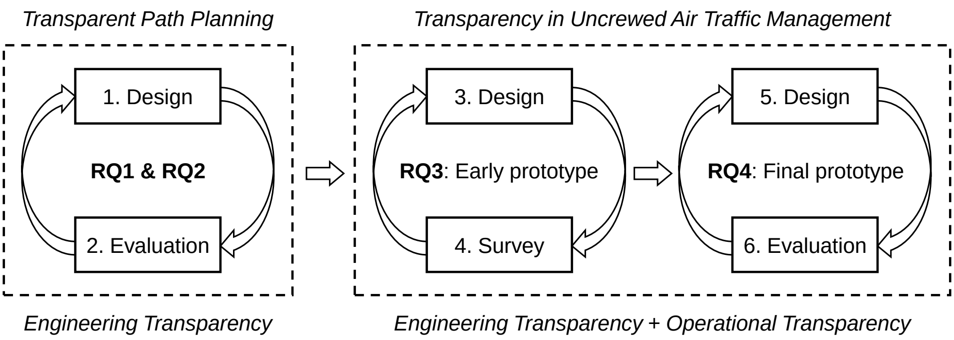


Figure 6.1: Design cycle in this dissertation.

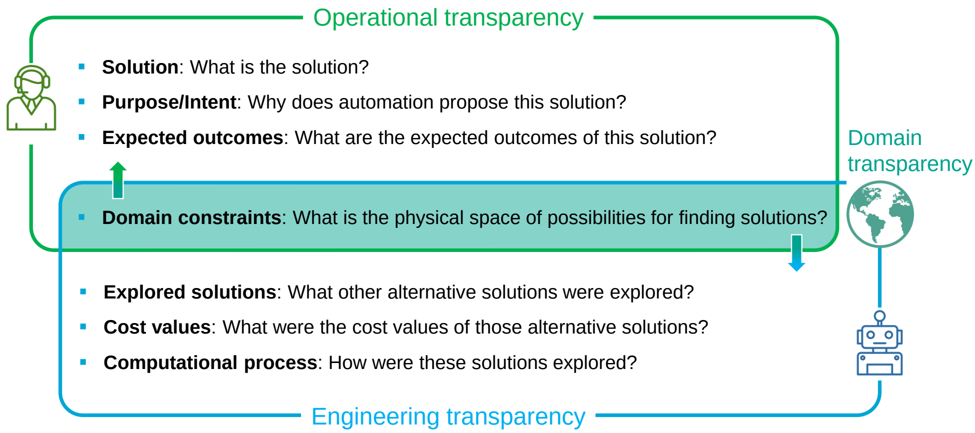


Figure 6.2: Product 1: Proposed unified taxonomy for algorithmic transparency.

transparency elements for UTM, along with their early visual prototypes, were devised. A survey study was conducted to investigate operators' opinions on the usefulness of these elements for supervising UTM operations.

Using the feedback received from the survey, a final interface prototype was developed within the third iteration, as depicted in Figure 6.3. To evaluate this interface, a human-in-the-loop experiment was conducted to explore the usage of transparency in real-time, tactical UTM operations (Chapter 5).

6.1.1. TREE-BASED VISUALISATION

Research Question 1	**Chapter 2**
How can the internal processes of path-planning algorithms be visualised via a unified approach, and what impact does such visualisation have on algorithm runtime?	

Given the focus on graph-based and sampling-based algorithms, the path-planning

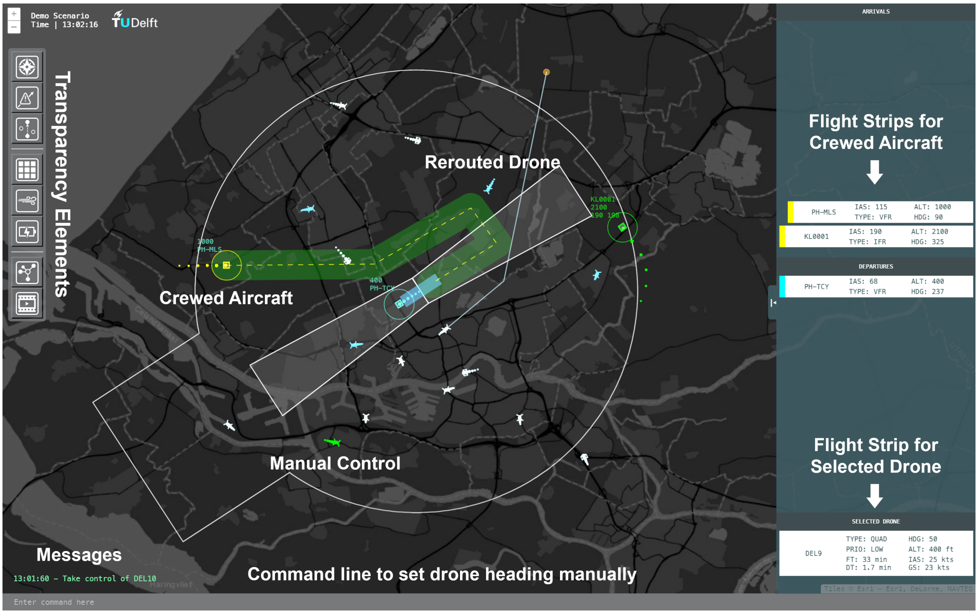


Figure 6.3: Product 2: UTM interface prototype DroneCTR[1].

process can be depicted by the growth of a tree from a start point to a target point. This tree, known as a search tree, may have unnecessary nodes pruned during its growth to conserve computing resources for efficiency (e.g., speedup and memory). The branches of the search tree, representing the connections between nodes, may also be adjusted to improve the tree structure for optimality (e.g., path length). Therefore, a visual approach to transparent path planning requires the storage, extraction, and visualisation of search trees throughout the planning process. A novel web-based pathfinding visualiser[2] was developed accordingly, incorporating more than ten representative path-planning algorithms. Recall that search nodes are not limited to points and can also take other forms, like triangles in Anya [65] and Polyanya [68]. The search tree thus has a broader interpretation. Overall, the proposed tree-based visualisation is applicable to a wide range of tree-based path-planning algorithms, primarily including graph-based and sampling-based ones.

For other types of algorithms, such as potential field-based [243] and reinforcement learning-based path planning [244], there is no search process and search tree. Potential field-based path planning treats obstacles as sources of repulsion and the target point as a source of attraction, thereby guiding the generation of a path towards the target. Instead of explicitly searching for a path, this approach constructs an artificial potential field. In this field, obstacles resemble mountains, while the target appears as a sunken valley. An agent can be regarded as a ball: no matter where it starts, it will eventually roll down and settle into the target area. However, a key drawback of this approach is its susceptibility

[1]URL: `http://dronectr.tudelft.nl/`, ID: transparency
[2]URL: `http://dronectr.tudelft.nl/`, ID: pathfinder

to becoming trapped in local minima, where the agent may be unable to reach the goal. Reinforcement learning-based path planning learns a policy to navigate an agent to its target. The policy specifies the action the agent should take based on its current state. Since the state mainly refers to the agent's position in path planning, the policy should ideally be capable of generating a feasible path from any start point. Therefore, the policy can also be seen as a field, with the agent behaving like a ball that rolls towards the target, regardless of its initial position.

It seems that the tree-based visualisation is not feasible for these types of path planning. However, since a search tree represents the connections between nodes, a tree-like structure can still be built to explicitly portray the artificial potential field or learned policy. As demonstrated in Figure 6.4, given any start point on the grid, the field or policy can guide the agent to the target point (red circle) via a path (orange line). By traversing all possible start points, a tree can be formed with the target point as its root. Essentially, the tree-based visualisation is intended to make the underlying navigation strategy of the path-planning algorithm visually salient – whether the strategy is predefined (potential field-based), learned (learning-based), or searched (graph- and sampling-based).

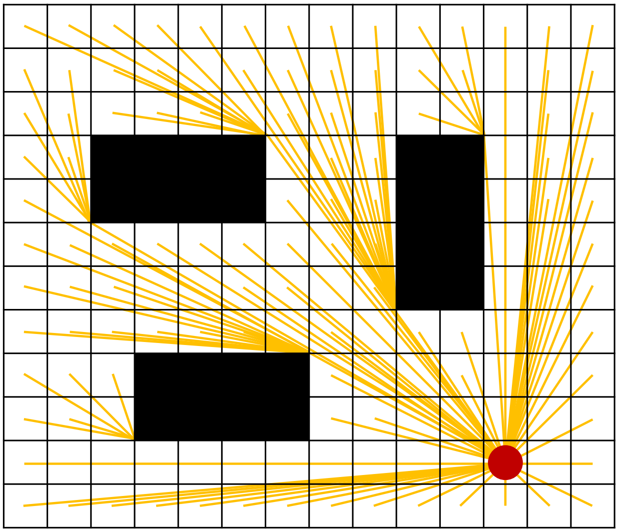


Figure 6.4: A schematic illustration of tree-based visualisations for potential fields or learned policies. The red circle is the target point.

As noted earlier, transparent path planning requires extracting information from the search process, which inevitably introduces additional computational overhead. Benchmarking results show that extracting all search trees representing the entire search process may significantly slow down the original path planning, with the degree of slowdown varying across different algorithms. To mitigate this impact, the search tree extraction can be performed afterwards or on a separate thread, rather than during the search process.

The impact of transparency on algorithm runtime could be a concern for real-time, large-scale operations, particularly those involving large maps. In this case, human operators may encounter delayed feedback from the algorithm, leading to slower responses, mistimed corrections, and ultimately unstable human-machine interactions. To address this issue, one approach is to use a faster path-planning algorithm that expands only the necessary branches, pruning unnecessary ones early in the search process. This can improve both path planning and information extraction speeds. Alternatively, the system's computational load can be reduced by lowering the level of transparency (RQ2), thereby limiting the amount of information that needs to be extracted and revealed.

For algorithm designers and policymakers, this slowdown may be of little concern. In fact, a slower algorithm might even improve their evaluation of the algorithm [105]. The additional waiting time allows them to reflect on the problems and potential solutions, thereby deepening their understanding of the algorithm's decision-making process (e.g., for debugging and auditing).

6.1.2. IMPACT OF TRANSPARENCY ON UNDERSTANDING

Research Question 2	**Chapter 3**
How can visual algorithmic transparency information in path planning be organised in a structured way, and how does that affect human understanding?	

6

The primary objective of algorithmic transparency is to enhance human understanding of how an algorithm functions, and thus transparency should be designed with human users in mind. To avoid overwhelming users, transparency is generally structured into hierarchical levels, facilitating the progressive disclosure of the algorithm's internal mechanisms [22, 30]. Six levels of transparency for path planning were proposed in Chapter 3. In these levels, the path-planning information was chunked into distinct elements, such as search space, discretised space (predefined or random graphs), explored nodes, and search trees. These elements were organised such that more detailed information can be presented at higher levels of transparency. The final level involves animating the search process, corresponding to Chapter 2 (RQ1).

To assess the effectiveness of algorithmic transparency in path planning, a user study was conducted to explore how human understanding evolves with the increasing levels of transparency provided. To enhance the generalisability of the findings and investigate whether the type of algorithm influences understanding, two distinct path-planning algorithms were chosen: Theta* (graph-based) and Informed RRT* (sampling-based). The experiment results indicate that increased transparency enables non-experts to understand path-planning algorithms more confidently and accurately, with sampling-based algorithms likely being easier to comprehend than graph-based ones.

While the overall trend in the experiment suggests that increased transparency can enhance understanding, it is also found that this effect may not always hold, particularly when the algorithm's behaviour violates human expectations [30, 148]. For example, when transparency was low, some participants believed that Theta* generated its search nodes at the corners of obstacles, as the paths it found were relatively straight and the turning points often appeared around obstacle corners. However, with increased transparency, this expectation was challenged when it became evident that Theta* in fact explored the search space grid by grid. This contradiction resulted in some degree of confusion. For Informed RRT*, the incomplete overlap between its sampling points and explored nodes also confused some participants upon their first exposure to the explored nodes (Level 3). They initially thought that the sampling points reflected the space and locations explored by the algorithm. However, actually, these points were used only to discretise the search space and guide the exploration direction of the search tree.

This finding aligns with the concept of model reconciliation in Explainable AI Planning (XAIP) [73, 74], where the differences between human mental models and AI models are considered the cause of the need for explanations. Increasing transparency can also be regarded as a means to correct human mental models, which helps explain why greater transparency often leads to improved understanding. However, the visual transparency designed in this research is not tailored to specific individuals. Participants were required to interpret the presented information on their own. To further enhance understanding and reduce confusion, (interactive) textual or verbal explanations could be incorporated to clarify the elements and processes involved in path-planning algorithms. These explanations could more precisely convey the semantic content of the visual information.

The study results also show that sampling-based path-planning algorithms may be easier to understand than graph-based algorithms. One possible reason is that the random exploration strategy of Informed RRT* is readily observable. For example, Informed RRT* will generate different paths in the same environments. Even with minimal transparency (Level 1), most participants were able to recognise that the algorithm employed a certain random approach. Although the randomness of Informed RRT* makes its next step during exploration unpredictable, the overall pattern is clear: the search expands randomly from the start point and is limited within a certain region (the search space). In contrast, Theta* relies on deterministic procedures, with embedded line-of-sight checks that enable it to ignore adjacent grids and construct more direct paths. This strategy will cause a mismatch between the edges of the graph (adjacent grids) and the branches of the search tree (paths), making the algorithm's behaviour less intuitive to interpret.

Certainly, the difficulty of understanding an algorithm also depends on its complexity. Theta* cannot represent the entire family of graph-based path-planning algorithms. A*, which usually follows the edges of the graph exactly, may be easier to comprehend than Theta*. Polyanya, with its unconventional graph structure and search nodes, might be more challenging to understand. These speculations warrant further experimental investigation. Contrary to the expectation, the study overall suggests that the random strategy used in sampling-based path planning is generally meaningful and intuitive to humans, and may be more easily understood than approaches based on predefined graphs.

6.1.3. UNIFIED TRANSPARENCY TAXONOMY

Research Question 3	**Chapter 4**
What constitutes algorithmic transparency in tactical UTM operations, and what are operators' perceptions regarding its role in supporting UTM supervision?	

In tactical UTM operations, the goal of UTM is to safely navigate drones to their destinations. This is essentially a (conflict-free) path-planning problem. Therefore, Chapter 4 builds upon the research on transparent path planning in Chapters 2 and 3, extending it to the UTM context. Here, algorithmic transparency encompasses not only the internal processes of path-planning algorithms, but also their interactions with the surrounding environment and operational context.

A unified taxonomy was therefore proposed, integrating operational, domain and engineering transparency, as presented in Figure 6.2. The taxonomy is structured around the "solution" proposed by an algorithm, with each category reflecting a distinct aspect of transparency. The operational transparency can be seen as a variant of the Situation Awareness-based Agent Transparency (SAT) model [22], and the proposed three categories align with the SAT model's three levels. The domain transparency is derived from Ecological Interface Design (EID) [53] and Cognitive Work Analysis (CWA) [57], both of which emphasise the importance of making the underlying structure of the work domain visually salient. The engineering transparency is inspired by Explainable AI (XAI) [31] and transparent path planning. The three proposed categories of engineering transparency also correspond to the three categories of operational transparency.

Based on the proposed taxonomy, twenty transparency elements and fourteen corresponding visual prototypes for supporting UTM supervision were devised in Chapter 4. A survey-based user study was conducted to explore operators' needs and preferences regarding these elements and prototypes, and to validate the effectiveness of the proposed taxonomy. Since UTM is not yet operational, no dedicated UTM operator or supervisor is available. Thus, twelve professional ATCos and twelve drone experts, whose roles are closely related to UTM, were invited to participate in this study.

Results show that operational transparency is preferable to engineering transparency in nominal UTM operations. In UTM failure scenarios, both types of transparency were considered useful and valuable. The study also asked participants to group the proposed twenty transparency elements, and their groupings closely aligned with the proposed taxonomy. Additionally, no significant differences were found between ATCos and drone experts regarding their transparency needs and preferences, suggesting that a "one-size-fits-all" transparency solution for UTM would be possible.

When the focus shifted from transparent path planning to UTM routing, it was surprising to find that the AI and ATM communities seemed to have different interpretations of transparency. Research on transparency in ATM emphasises operational transparency that supports operators in maintaining situation awareness when collaborating with automation [55]. In contrast, XAI seeks to reveal the inner workings of AI to make its functioning understandable to humans [32, 33], which is referred to as engineering transparency in this research. Endsley [70] also discussed the distinction between these two perspectives (but using different terminology). In this study, it is found that operational transparency is favoured over engineering transparency in nominal UTM operations.

Essentially, this difference in focus on transparency stems from the goals and needs of users. In operational contexts, operators are mainly concerned with completing their tasks safely, efficiently and/or sustainably. They may not need to understand how the algorithm arrived at a solution, but rather how that solution affects the overall situation and environment. Operational transparency helps operators assess whether a proposed solution aligns with their goals, and whether an alternative solution or manual intervention is required. In contrast, in the AI field, opening the "black box" is usually necessary to develop trustworthy systems and facilitate their real-world applications. Engineering transparency helps developers debug their software and enables policymakers to audit algorithmic systems. The survey study reveals no significant difference between ATCos and drone experts, despite their different professional backgrounds and experiences. This can also be attributed to the fact that they were assigned the same role (UTM supervisor), and thus their transparency needs were similar.

However, this does not imply that engineering transparency is useless in operational contexts. When automation fails or exhibits unexpected behaviour, operators may need engineering transparency to understand what happened and why it happened [13]. This is also reflected in the survey study, where participants expressed a desire to access all relevant transparency information in automation failure scenarios.

However, the relevance of the information and the manner in which it is presented are contingent on the particular context and circumstances. Therefore, a more systematic approach is needed to determine what is relevant and under which conditions. The Joint Control Framework (JCF) may serve this purpose, complementing the proposed transparency taxonomy with an analysis of human-automation interaction patterns [245]. The JCF provides a framework for analysing interaction patterns between humans and automation at different levels of cognitive control, modes of control allocation and possible behavioural trajectories over time. Depending on the interaction pattern, the focus of transparency may shift. For instance, when automation operates at the execution level, humans may only require certain elements of operational transparency (e.g., expected outcomes). Conversely, when automation shifts towards higher-level planning under time pressure, a small subset of engineering transparency elements may be required to reveal problems in the underlying computational and/or reasoning processes.

6.1.4. USAGE OF TRANSPARENCY FOR UTM SUPERVISION

Research Question 4	**Chapter 5**
How do operators use algorithmic transparency in tactical UTM supervision?	

Based on insights from the survey-based user study (RQ3), the transparency elements proposed in Chapter 4 were further improved and implemented as eight buttons in the UTM interface, as presented in the top left of Figure 6.3. In the survey study (RQ3), the responses provided by operators were based on their previous operational experience. Without direct exposure to transparency in real-time operations, their transparency needs reflected in the survey may not accurately represent their actual requirements. Therefore, based on the UTM interface, a human-in-the-loop experiment was conducted to explore how operators use the transparency elements in practice. In alignment with Chapter 3 (RQ2), a comparison was also made between a graph-based path-planning algorithm, Zeta*-SIPP, and a sampling-based algorithm, Informed RRT*, in supervising UTM operations. In this case, the path-planning algorithm is capable of avoiding not only static obstacles (e.g., geofences) but also dynamic obstacles (e.g., crewed aircraft).

Results show that information regarding the Closest Point of Approach (CPA) between drones and crewed aircraft ("Expected Outcomes" category in the proposed taxonomy) is particularly useful for supporting UTM supervision. Participants proactively and frequently accessed the CPA information on drone flying routes to ensure there were no potential conflicts with crewed aircraft. The CPA prediction for drone direct routes was also often used to determine whether drones could fly directly to their destinations to complete their missions or reduce flight delays.

The frequent use of the CPA information is probably due to the difficulty in predicting the minimum separation between drones and crewed aircraft. This challenge is caused by several factors, such as the large speed difference between drones and crewed aircraft, the substantial impact of wind on drone speed, the varying speeds of takeoff and landing flights, and the complex trajectories of VFR (Visual Flight Rules) flights. The CPA predictions in UTM differ significantly from those in traditional ATC, where aircraft typically operate at similar speeds. Thus, even for professional ATCos, making such CPA predictions in UTM is still difficult without proper long-term training tailored to UTM supervision [14, 63]. The CPA information can help reduce operators' efforts and support them in supervising UTM operations. In fact, CPA prediction or conflict detection is also central to traditional ATC and forms the basis for many ATC decision-support tools [54, 187, 195].

This finding suggests that, to effectively support human supervision, automated systems should be designed to facilitate human evaluation of solutions proposed by algorithms. The situational changes, impacts, and consequences that the proposed solutions may cause should be clearly communicated to users, particularly when these outcomes are difficult for humans to predict. Therefore, operational transparency can be regarded as a means to support human evaluation of situations and proposed solutions. Similarly, engineering transparency can be seen as a way to support the evaluation of the underlying procedures and processes. For instance, it enables users to assess whether an automated system operates as intended and, if not, to identify the cause of the malfunction.

6

While the transparent UTM interface was specifically designed for supporting supervision, the experiment findings also underscore the importance of supporting intervention. In human-automation collaboration, transparency allows humans to understand what the automation is doing and examine whether it is functioning as expected. However, if something goes wrong, the interface should also provide the means for humans to intervene and assume control. In such situations, offering decision-support tools may be even more necessary than in non-automated settings, to reduce the sudden frustration and pressure caused by automation failures or unanticipated behaviour [169]. Otherwise, operators may become overwhelmed, resulting in distraction and a diminished awareness of the overall situation, as well as reduced trust and acceptance.

In Chapter 3 (RQ2), it was found that the sampling-based path-planning algorithm Informed RRT* is easier to understand than the graph-based algorithm Theta*. In Chapter 5 (RQ4), it was also observed that Informed RRT* may be more suitable for UTM supervision than the graph-based algorithm Zeta*-SIPP. Zeta*-SIPP can be regarded as an enhanced version of Theta*, capable of finding *time-optimal* any-angle paths in environments with *dynamic obstacles.* The experimental results indicate that Informed RRT* generally results in a lower workload, especially in the "complex-failure" scenarios. This is likely because the explored space in Informed RRT* was more intuitive to participants, particularly the notion that each branch of the search tree corresponded to an exploratory and feasible action. As a result, the structure of its explored space could offer visual cues about which directions were more likely to resolve conflicts and which were not.

However, this research cannot conclusively recommend that UTM should employ a sampling-based path-planning algorithm such as Informed RRT*. Some participants explicitly expressed their dislike for the randomness of Informed RRT* and voiced concerns about its reliability and optimality. The advantages of Informed RRT* over Theta* and Zeta*-SIPP, identified in the experiments (RQ2 and RQ4), could also potentially be offset through sufficient pre-training for understanding, improved visualisation for grid-based path planning, and the development of decision-support tools for human intervention.

Regardless of the algorithms implemented in automated systems, operational users should undergo thorough training before actual operation. Transparency can play a crucial role in supporting this process, and users should also be properly trained in utilising transparency elements effectively. Overall, selecting the "best" algorithm for a human-centred automated system should take into account multiple factors, including the algorithm's optimality, efficiency, reliability, transparency, and interpretability.

6.2. RECOMMENDATIONS

In this dissertation, some assumptions were made to define and limit the research scope. However, these assumptions may not always hold in real-world applications. Therefore, several recommendations for future exploration are discussed as follows.

6.2.1. EXTENSIONS TO LEARNING-BASED ALGORITHMS

This study focuses on graph- and sampling-based path-planning algorithms and, based on this foundation, proposes a unified taxonomy for algorithmic transparency. However, owing to the substantial potential offered by machine learning, future research could further explore how to extend the taxonomy to include machine learning approaches as well, addressing transparency for training data, training algorithms and trained models [33]. For instance, visualising the policy in reinforcement learning could offer deeper insights into the AI decision-making strategy [205]. Revealing the learning process could also assist policymakers in identifying bias in learning-based AI models [28].

Figure 6.5 presents a possible extension of the proposed transparency taxonomy to incorporate machine learning algorithms. Since operational transparency mainly concerns the proposed solution and its interaction with the external environment, the distinction between heuristic/optimisation and machine learning algorithms lies in (internal) engineering transparency. Each type of engineering transparency in machine learning algorithms has a direct counterpart in heuristic/optimisation algorithms. Explanations for trained models or policies in machine learning are also associated with XAI. Further research is required to validate and refine this extended transparency taxonomy.

For graph- and sampling-based algorithms, a visual approach is proposed to achieve transparent path planning based on search trees (Chapter 2). Essentially, this tree-based visualisation is founded on the fact that the algorithm explores the space from the start point until the target point is reached. The explored paths construct a tree-like structure, making them particularly suitable for tree-based visualisation. The end of a search tree branch means that a feasible path from the start point to that endpoint has been found. For an A*-based algorithm, when a node is closed, it is considered that the optimal path to that node has been identified.

As previously discussed, the tree-based visualisation can also be applied to potential

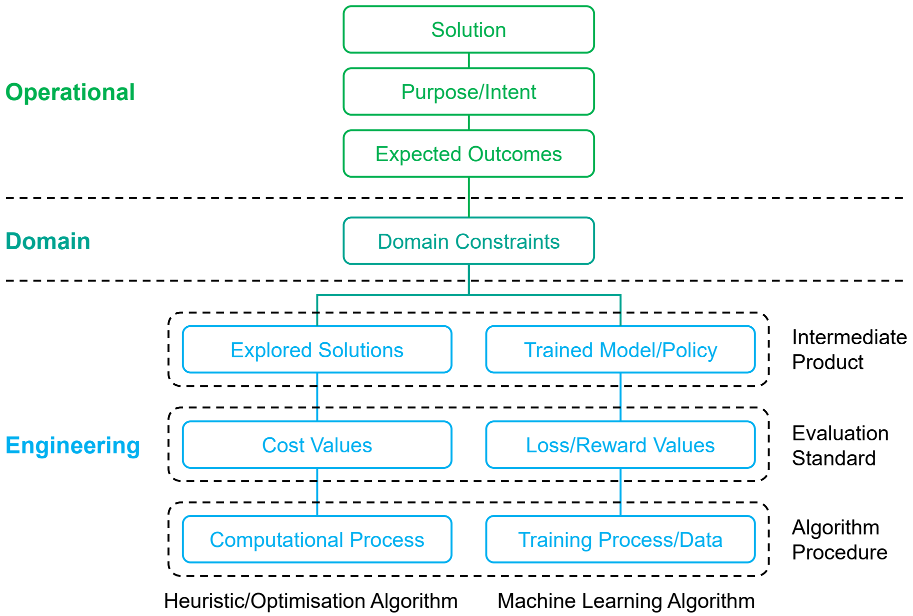


Figure 6.5: Extended transparency taxonomy including machine learning algorithms.

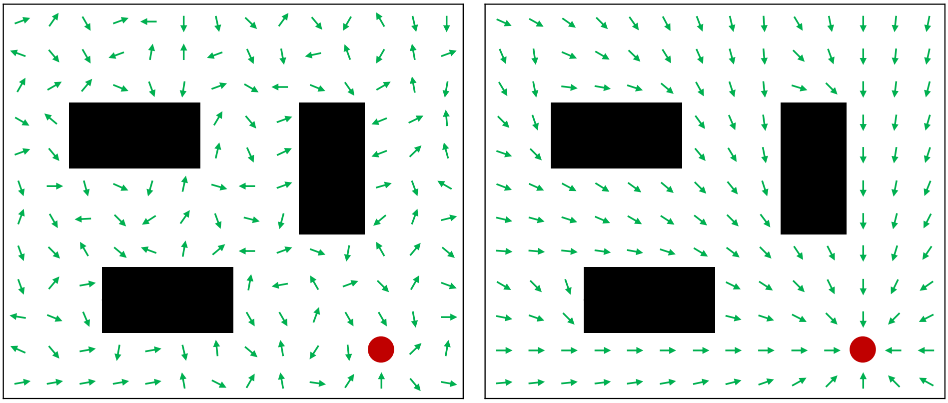
field-based and reinforcement learning-based path planning, where it serves to indicate the artificial potential field or learned policy. However, before the policy is fully learned or finalised, a clear tree structure may not emerge, resulting in a chaotic or unstructured visualisation. Therefore, to portray the training process, a vector field–based visualisation may be more appropriate [246]. Figure 6.6 illustrates an example of visualising the initial and final policies based on vector fields. The training process can be viewed as analogous to the search process. If all intermediate policies during training were visualised, one might observe the vectors gradually converging towards the target (red circle). Potential field-based path planning can be viewed as a special case, with the algorithms being "trained" manually by designers.

Beyond pure reinforcement learning-based path planning, some other learning approaches have been proposed that can learn heuristics for A*-based algorithms [247] or sampling distributions for RRT*-based algorithms [248]. This type of learning-based algorithm can also be visualised using search trees, as its learning modules, like plug-ins, do not alter the core procedures of graph-based and sampling-based algorithms.

6.2.2. ADAPTABLE AND ADAPTIVE TRANSPARENCY

In this dissertation, adaptable transparency is implemented, allowing users to freely activate or deactivate transparency elements themselves. This approach contrasts with adaptive transparency, where changes in transparency information are governed by automation. Both approaches have their own advantages and disadvantages.

Adaptable transparency offers users full control over the display of transparency in-



(a) Random initialisation (b) Final policy

Figure 6.6: Vector field-based visualisation for reinforcement learning policies.

formation, helping to prevent automation surprises caused by unexpected automation-driven changes [171, 234]. By enabling manual adjustments, it can enhance system flexibility, support task-specific customisation, and foster a stronger sense of user agency. However, this approach introduces some challenges as well. For example, providing too many control options could overwhelm users, increasing their cognitive load and diverting their attention from primary tasks. There is also a risk that users may configure transparency settings suboptimally, potentially impairing display visibility and reducing overall task performance.

In contrast, adaptive transparency dynamically adjusts transparency information according to automation-driven assessments of the user's context, workload, and task demands. By automating the management of information presentation, it can reduce user effort, maintain an optimal balance between information richness and cognitive load, and support smoother task execution, especially in high-demand situations. However, users may struggle to develop accurate mental models of when and how transparency information is presented. Transparency cues may sometimes appear, disappear, or change unexpectedly, potentially distracting users and disrupting their situation awareness.

Combining adaptable and adaptive transparency (hybrid) may offer the most effective strategy for supporting user needs. Adaptable transparency can be applied during routine operations, allowing users to customise their experience as needed. In contrast, adaptive transparency mechanisms can be activated in urgent situations, such as conflicts, loss of separation, or emergencies, to ensure that critical information is delivered promptly without overwhelming users [249]. Future research could explore methods for achieving adaptive transparency and systematically compare adaptable, adaptive, and hybrid transparency mechanisms within the context of UTM (or ATM).

6.2.3. CENTRALISED AND DISTRIBUTED UTM

This research adopted a centralised UTM operational concept within controlled airspace, aligned with the U-space framework. This approach mirrors existing ATM systems, thus facilitating smooth integration of UTM into ATM operations. Under this centralised concept, a single (path-planning) algorithm is applied to manage all drones, making it easier for human operators to supervise the UTM operations. Centralised UTM can enable global planning, thereby improving overall safety and efficiency. It can also enhance the security of ATM by preventing drones from directly accessing information about crewed aircraft. However, centralised UTM relies heavily on the capability of the implemented (path-planning) algorithm. As the number of drones increases significantly, the centralised algorithm may encounter scalability limitations, leading to delays in response time.

In addition to centralised UTM, a distributed UTM approach is also possible [8] and may help to overcome the shortcomings of the centralised concept. Centralised UTM can be regarded as a single "mind" governing all drones, whereas in distributed UTM, each drone possesses its own "mind". This allows distributed UTM to scale more effectively, managing more drones with less concern about time and memory complexity [207]. In this distributed concept, the safety responsibilities of drone operations are assigned to individual drones or their operators rather than to a centralised system or its supervisor. In this case, the role of UTM services is limited to providing non-mandatory advisories and sharing intent information among aircraft. The final decision is made by either autonomous drones or by drone operators. However, without a centralised UTM system, coordination between ATM and UTM may be more difficult, which could potentially be problematic in controlled airspace.

Therefore, a hybrid UTM system may be the most effective approach to balance centralised oversight with decentralised operational flexibility. For example, the centralised approach can be employed in controlled airspace where drone operations intersect with crewed aviation, ensuring coordination and safety. In contrast, distributed management can be more suitable for rural and urban environments dominated by drone operations, where scalability and responsiveness are critical.

Nevertheless, the proposed approach for achieving algorithmic transparency is applicable to both centralised and distributed UTM, as long as human supervision of drone operations is required. In centralised UTM, the human supervisors are UTM operators, similar to ATCos, whereas in distributed UTM, the supervisors are drone operators. The application scenarios will not affect the main findings of this research, although in distributed UTM, the supervisors are only responsible for monitoring their own drone operations rather than the entire airspace.

6.2.4. GEOFENCING AND GEOCAGING

Geofencing is used to define no-fly zones for drones. This research implemented grid-based geofences based on previous research on ATM–UTM collaborative operations [13, 14, 63]. This design choice was made because fixed grids may be easier for humans to understand and manipulate, and regular grid structures help limit display complexity.

There are other types of geofences as well, such as circular and polygonal geofences [250]. Rather than relying solely on activating fixed grids, perhaps UTM could allow operators to define geofences by drawing custom polygons. This design can limit the number of geofences (compared to grids) and make it easier to adjust their parameters, such as shape and time dimension. However, a potential drawback is that the resulting airspace structure may become complex and inconsistent across different operators. Further research is required to determine which type of geofences is most effective for UTM.

In addition to geofencing, geocaging has been proposed to restrict drones to operate within a certain area [11]. Geocaging is particularly suitable for fully segregating drones from crewed aircraft, limiting interactions and potential conflicts between them. However, such interactions cannot be entirely eliminated, and geofencing could play a role in managing them. When a crewed aircraft needs to access a geocaged airspace, geofences can be dynamically added to separate drones from the crewed aircraft. Similarly, when a drone has to cross a region where crewed aircraft are taking off or landing, geofences can also be utilised to protect crewed aircraft from potential drone incursions.

Essentially, geofencing is to define ATM airspace for crewed aircraft, while geocaging is to define UTM airspace for drone operations. These two techniques can be applied to the AURA scenarios for ATM–UTM collaborative operations, as shown in Figure 1.2 in Chapter 1. Future research could further explore how to combine these two techniques to support Dynamic Airspace Reconfiguration (DAR) in high-density operations.

Additionally, in the experiment in Chapter 5, participants actively controlled drones instead of relying on geofences, probably because the number of drones was low and relatively manageable through manual control. While the number of drones is expected to increase significantly in the future, the interactions between drones and crewed aircraft should be limited to minimise risk and potential conflicts. It can be considered that irrelevant drones were automatically filtered out using geocaging and were therefore not displayed during the experiment. As a result, participants only needed to supervise drones that might have an issue, with both geofencing and manual control available. Future research could introduce a more realistic operational environment and allow operators to configure both geofencing and geocaging settings, in order to further explore which control methods are most preferred and frequently used for UTM.

6.3. CONCLUSIONS

This dissertation shows how to achieve algorithmic transparency for UTM with a focus on path-planning algorithms. The goal of algorithmic transparency is to make the inner workings of an algorithm accessible and comprehensible to humans. The experiment in this dissertation confirms that increased transparency can enhance human understanding, and the survey study indicates that both ATCos and drone experts consider transparency essential for UTM supervision. Algorithmic transparency can help humans better understand an algorithm's capabilities and limitations, preventing blind acceptance or rejection of its decisions and fostering well-calibrated trust.

However, the need for transparency is not fixed. As indicated in the survey study and the final human-in-the-loop experiment, it may vary across different scenarios and contexts. In nominal scenarios, operators typically require less transparency, whereas in failure scenarios, their need for transparency increases. Operators' transparency needs are also affected by their familiarity with automation. A deep understanding of the automation's functionality can lessen their reliance on transparency. When automation becomes fully safe, reliable and trustworthy, operators may no longer need to seek transparency – except occasionally out of curiosity. For example, one can use a mobile phone effectively without possessing detailed knowledge of its internal mechanisms.

Algorithmic transparency often involves storing, extracting and presenting information from algorithms, at the expense of increased computational resources. For real-time operations with limited resources, a trade-off may be required between the amount of information revealed and system performance. Additionally, to avoid overwhelming users with information, transparency is typically organised into distinct levels, with more details progressively disclosed. Designing for transparency should take into account both human (cognitive load) and machine (computational resources) limitations.

Algorithmic transparency can take various forms, including visual, textual, and verbal representations. This research primarily focuses on visual transparency, which excels at making complex (spatiotemporal) relationships more visually salient. However, visual transparency may be less effective at conveying precise rules, abstract concepts, and numerical comparisons, where textual explanations often perform better. Likewise, it lacks the natural, conversational quality of verbal communication, which can be more suitable in scenarios where users' eyes are occupied and visual input cannot be effectively processed, such as during driving or performing surgery. To enhance human-computer interaction, a multimodal approach that integrates multiple forms of transparency is recommended. In path planning, visualising each step of the algorithm alongside the corresponding line of code being executed can probably further enhance understanding.

Overall, transparency is an important step towards trustworthy automation, acting as a communication bridge between humans and automated systems. However, trustworthiness also depends on the automation itself being reliable, reasonable, and responsible. Humans, automation, and their interactions all play vital roles in achieving this goal. As long as automation still requires human supervision, transparency is indispensable.

A

ZETA*-SIPP

Zeta-SIPP is a state-of-the-art grid-based path-planning algorithm designed for dynamic environments where the trajectories of moving obstacles are predictable. This is an extension to any-angle path planning that is particularly suitable for unstructured or flexible spaces, such as airspace, warehouses, and oceans. It has been applied to drone rerouting in the UTM simulator and experiments (Chapters 4 and 5). By achieving transparent path planning to enhance non-experts' understanding, I also gained deeper insights into path-planning algorithms. This, in turn, enabled me to develop this novel algorithm Zeta*-SIPP.*

The contents of this chapter are based on:

Paper title	Zeta*-SIPP: Improved time-optimal any-angle safe-interval path planning
Authors	Yiyuan Zou and Clark Borst
Published in	33rd International Joint Conference on Artificial Intelligence (IJCAI) 2024
DOI	10.24963/ijcai.2024/754

ABSTRACT

Any-angle path planning is an extension of traditional path-planning algorithms that aims to generate smoother and shorter paths in graphs by allowing any-angle moves between vertices, rather than being restricted by edges. Many any-angle path-planning algorithms have been proposed, such as Theta, Block A* and Anya, but most of them are designed only for static environments, which is not applicable when dynamic obstacles are present. Time-Optimal Any-Angle Safe-Interval Path Planning (TO-AA-SIPP) was developed to fill this gap, which can find an optimal collision-free any-angle path that minimises the traversal time. However, as indicated by its authors, TO-AA-SIPP may not be efficient enough to be used in Multi-Agent Path Finding (MAPF). Therefore, this chapter presents a new algorithm Zeta*-SIPP to improve TO-AA-SIPP by means of 1) reducing useless search nodes that have no contribution to finding optimal solutions, and 2) introducing Field of View (FoV) instead of Line of Sight (LoS) to speed up visibility checks with static obstacles. Benchmark experiments showed that Zeta*-SIPP reduced the computation time of TO-AA-SIPP by around 70%-90% on average.*

A.1. INTRODUCTION

Path planning aims to find an optimal path between two locations. A* [50] is one of the most classic algorithms to solve the path-planning problem. However, the paths found by A* are usually not the true shortest because the expansion of A* is limited to adjacent neighbours and thus the shape of the path is highly affected by the graph structure. On a square grid map, A* only searches paths in 45-degree increments. Therefore, any-angle path planning has been developed, which ignores the edges of the graph and allows any-angle turns at vertices, to generate smoother and shorter paths, such as Theta* [66], Block A* [87] and Anya [65]. More analysis and evaluation regarding any-angle path planning can be found in [251]

Most any-angle path-planning algorithms are designed only for static environments, limiting their applicability in the presence of dynamic obstacles. Path planning with dynamic obstacles needs to consider the time dimension when searching for collision-free paths. A common approach to handle it is to divide the time dimension into multiple equal-length time slots, thus creating a space-time grid map [252]. In this way, static and dynamic obstacles can both be represented by space-time grids, and thus path-planning algorithms, like A*, can be easily extended to dynamic environments. However, this approach may aggravate the *curse of dimensionality* problem because obstacle grids could be quite sparse, especially when there are only few dynamic obstacles. Therefore, Safe Interval Path Planning (SIPP) [88] was proposed to merge consecutive obstacle-free time slots into safe intervals, creating a *compact* space-time map to narrow the search space.

To combine any-angle path planning and SIPP, Any-Angle SIPP (AA-SIPP) was designed [253]. The basic idea of AA-SIPP is similar to Theta*. Both attempt to straighten paths by checking if the neighbours of a current node can be reached from the parent of this current node with lower cost. However, this is a greedy approach. AA-SIPP, like Theta*, is not guaranteed to find true optimal any-angle paths. Thus, Time-Optimal AA-SIPP with inverted expansion (TO-AA-SIPP is used in this chapter instead of iTO-AA-SIPP) was further developed [84] by checking if the paths through the current node to its neighbours could be straightened by any other node in the search space rather than only the parent of the current node. This operation ensures optimality but also slows down the algorithm since too many nodes need to be examined during the search process. As stated in [84], TO-AA-SIPP may not be fast enough to be straightforwardly used in Multi-Agent Path Finding (MAPF).

With further research, it was found that actually, in TO-AA-SIPP, not all nodes in the search space contributed to finding optimal paths. Some nodes were never removed from the $open$ list after being inserted during initialisation. Therefore, it should be possible to improve the performance of TO-AA-SIPP by reducing these useless nodes. Inspired by Informed Rapidly-exploring Random Tree* (Informed RRT*) [83] and Batch Informed Trees (BIT*) [90], the "ellipse" used in Informed RRT* and BIT* can be adapted to indicate and limit the *current* search range. Thus, an *any-angle forward expansion* was developed to incrementally add necessary nodes to the $open$ list, like the forward expansion in A*. In this case, nodes outside the "ellipse" are temporarily useless until they are expanded by this "ellipse" (search range). By implementing this idea, Zeta-SIPP was designed.

However, in the worst case, Zeta-SIPP would be as slow as TO-AA-SIPP because all nodes in the search space would be expanded by the any-angle forward expansion ("ellipse"). Therefore, to further improve TO-AA-SIPP and also Zeta-SIPP, *Field of View* (FoV) was introduced to replace Line of Sight (LoS) for collision detection (visibility checks) with static obstacles and thus developed TO-AA-FoV-SIPP and Zeta*-SIPP respectively. Zeta*-SIPP can be considered a superior version of Zeta-SIPP since their search processes are nearly identical except for the visibility check methods. According to the benchmark experiments, Zeta*-SIPP outperformed Zeta-SIPP and TO-AA-FoV-SIPP in most cases, especially when the map was large.

A.2. PROBLEM STATEMENT

Suppose there is an agent navigating from a start p_s to a target p_t in a graph $G = (V, E)$ where V is the set of vertices and E is the set of edges. Two different types of actions are allowed: *move* and *wait*. It means that the agent can *wait* at a certain vertex or *move* from a vertex to the other vertex. The movement speed is constant and the cost of an action is its duration. The agent can only turn or wait at the vertices and the inertial effects are neglected. Please note that in some practical cases, the wait action is not possible, such as fixed-wing aircraft. To simplify the problem, the radius of the agent is ignored.

A path plan is an ordered sequence of position-time pairs $\pi = \{(p_1, t_1), (p_2, t_2), ..., (p_n, t_n)\}$ where p_i represents a position and t_i denotes the waiting time at the position p_i. If the wait action is forbidden, $t_i = 0\,(i = 1, 2, ..., n)$. The cost of the path plan is the sum of the duration of the actions. The goal of this problem is to find the time-optimal plan from a given start p_s to a given target p_t.

It is assumed that the plans of dynamic obstacles are known: $\{\pi^1, \pi^2, ..., \pi^k\}$, and after the plans are accomplished, the dynamic obstacles will disappear. This is reasonable for flying vehicles. For instance, when drones complete their missions, they will land to reload or recharge. Some papers assume that the dynamic obstacles will stay in their target vertices forever [84] as on the ground, a robot cannot suddenly disappear and it will become a static obstacle after reaching its target.

A.3. ALGORITHM DESCRIPTION

A.3.1. OVERVIEW

Before diving into the details of the proposed algorithms, this section briefly introduces the basic idea of TO-AA-SIPP [84] and illustrates how it can be improved.

To make the explanation more easily understandable, the A* algorithm is introduced at first, as A* is quite well-known and also TO-AA-SIPP is A*-based. In A*, after a node is moved from $open$ to $closed$, eight neighbour nodes of this closed node are generated. If a neighbour is not visited before, this closed node will directly be its parent. However, if it has been visited, which means it already has a parent, A* needs to compare this closed node with the current parent of this neighbour node to find out which is better and could be its *true parent*. These two nodes can both be viewed as *potential parents* of this neighbour node, and the metric used for comparison is $g(pp(n)) + g(pp(n), n)$ where $pp(n)$ is the potential parent of a node n, $g(pp(n))$ is the real cost from the start to $pp(n)$ and $g(pp(n), n)$ is the real cost from $pp(n)$ to n. Therefore, the current parent of a node n is also its current *best potential parent* $bpp(n)$, and the real cost $g(n)$ can be written as

$$g(n) = g(bpp(n)) + g(bpp(n), n) \tag{A.1}$$

In TO-AA-SIPP, the cost computation is similar. However, to speed up the process, it delays the SIPP-based collision detection with *dynamic obstacles* until necessary. It means that TO-AA-SIPP utilises an estimated cost $h(bpp(n), n)$, the lower bound, to replace the real cost $g(bpp(n), n)$ in Eq. (A.1) when a node is inserted into $open$. After a node with the minimum cost is removed from $open$, the SIPP-based collision detection will be executed to compute the real cost $g(bpp(n), n)$ and determine if this node can be inserted into $closed$. This "lazy" evaluation is similar to Lazy Theta* [254] and BIT* [90] which delay line-of-sight checks with static obstacles. Please note that in TO-AA-SIPP, only the SIPP-based collision detection with dynamic obstacles is delayed while the line-of-sight checks with static obstacles are not. Hence, the cost function of TO-AA-SIPP is

$$\begin{aligned} f(n) &= g_{low}(n) + h(n) \\ &= g(bpp(n)) + h(bpp(n), n) + h(n) \end{aligned} \tag{A.2}$$

where $g_{low}(n)$ is the lower bound of $g(n)$ and $h(n)$ is the estimated cost from n to the target. This approach needs to record all the potential parents of the explored nodes. Then it can conduct the SIPP-based collision detection with dynamic obstacles later from the best potential parent to the worst potential parent until one can be the true parent (the detailed process is more complicated). Only when the true parent of a node is found can this node be inserted into $closed$ and meanwhile might be a potential parent of other nodes in $open$. Since this approach discovers and maintains connections between open nodes and their potential parents at each search step, it can be called *inverted expansion* [84], as shown in Figure A.1.

To find optimal any-angle paths, unlike the eight neighbours in A*, TO-AA-SIPP examines all nodes in the search space at each step. It means that in TO-AA-SIPP, when a node is closed, all the other nodes can be considered the "neighbours" of this closed node. The *extended "neighbours"* guarantee that TO-AA-SIPP is not limited to 45-degree

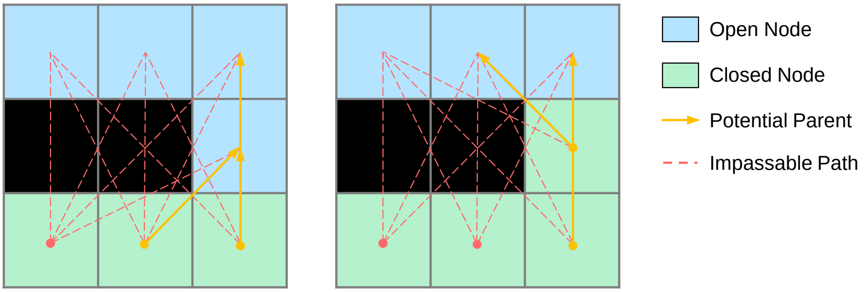


Figure A.1: Inverted expansion to find potential parents of open nodes.

increments and thus can find any-angle paths. However, not all search nodes may contribute to finding optimal paths. For example, if the map is large, it may be unnecessary to check the visibility connections between the closed nodes and the nodes that are very far from the start and target. In this case, many unnecessary line-of-sight checks may be conducted in TO-AA-SIPP since too many useless nodes exist in $open$, which reduces the performance of the algorithm.

To address this issue, a natural idea is to also delay line-of-sight checks, which can be performed with the SIPP-based collision detection simultaneously. In this way, only necessary line-of-sight checks will be executed. However, if there are many static obstacles, it may result in much more sorting computation in $open$. This is because after collision detection, $f(n)$ is updated. Only if $f(n)$ is still the minimum, the node n can be inserted into $closed$, otherwise, it has to be re-inserted into $open$ and reordered. There is a trade-off between collision detection and sorting computation when applying the "lazy" evaluation. Since the size of $open$ may be very large and grid-based line-of-sight checks can be implemented efficiently with line drawing algorithms, like Bresenham's line algorithm [255], Lazy TO-AA-SIPP could be much slower in the worst case. Therefore, this idea may not be suitable for improving TO-AA-SIPP.

Another idea is to directly reduce useless nodes in $open$, which will lead to both fewer line-of-sight checks and sorting calculations. In A*, forward expansion is applied to extend $open$ at each step, and the nodes in $open$ form a boundary of the A* search (search range). It is promising to develop a similar technique for TO-AA-SIPP to reduce the size of $open$ as in the worst case, only the calculations for forward expansion are redundant compared to the original TO-AA-SIPP. Therefore, to implement this idea, Zeta-SIPP is proposed and the details will be illustrated in section A.3.2.

Those familiar with computer graphics may recognise that line-of-sight checks may not be the best way to conduct collision detection with static obstacles in TO-AA-SIPP. The closed node can be viewed as a "light source" and the grids near the "light source" could be checked for multiple times, leading to a waste of computing resources. Therefore, *Field of View* (FoV) is introduced to replace line-of-sight checks. TO-AA-FoV-SIPP and Zeta*-SIPP are thus developed. Symmetric recursive shadowcasting [106] is applied to compute the Field of View efficiently.

A.3.2. ZETA-SIPP

The main idea of Zeta-SIPP is to reduce useless open nodes in TO-AA-SIPP using any-angle forward expansion. It means that only necessary nodes are expanded and inserted into *open* at each step instead of inserting all nodes in the search space into *open* at the beginning. Unlike A*, Zeta-SIPP aims to find any-angle paths, and thus the forward expansion of A* cannot be directly applied to Zeta-SIPP. Inspired by Informed RRT* and BIT*, an expanding "ellipse" is used to limit the range of any-angle search:

$$f_{low}(n) \leq \min_{m \in open} f(m) \quad \forall\, n \in S \tag{A.3}$$

where $f_{low}(n) = h(start, n) + h(n) \leq g(n) + h(n) = f(n)$ is the lower bound of $f(n)$ and S is the search space. The focal points of the "ellipse" are the start and target and the major axis length is $f_{low}(n)$. If all nodes satisfying the Inequality (A.3) are inserted into *open*, then

$$\min_{m \in open} f(m) < f_{low}(n) \leq f(n) \quad \forall\, n \in S_{out} \tag{A.4}$$

where $S_{out} = S \setminus (open \cup closed)$ is the search space outside the current search range. It indicates that the minimum cost in *open* is also the minimum in the remaining search space ($S \setminus closed$). Thus, at each step, only *open* is enough to find the next closed node, and there is no need to consider S_{out}. Therefore, the Inequality (A.3) can be applied to develop an any-angle forward expansion for Zeta-SIPP and the search range of TO-AA-SIPP can be limited during the search process.

The pseudocode of Zeta-SIPP is shown in Algorithms 5-9. To make the code structure clearer, the main loop is split into several modules, as shown in Algorithm 5. The revised part, compared to the original TO-AA-SIPP, is marked in red. The `findNextClosedNode` is to identify the next closed node n from the *open* list. Since the SIPP-based collision detection is delayed, the current node may be re-inserted into the *open* list after SIPP-based examinations (i.e., `validateTransition` in Algorithm 6). In this case, the next closed node would be *null*. The `invertedExpansion` assigns the closed node n as a potential parent to all nodes in the current *open* list, provided there are no static obstacles blocking the connection (i.e., `lineOfSight` in Algorithm 9). Please note that if the `invertedExpansion` is removed, Algorithm 5 can also represent the main loop of A*. The `forwardExpansion` utilises an expanding "ellipse" to continuously add new nodes into the *open* list. Here, a new list called *bound* is introduced to indicate the nodes around the boundary of the search range (*open* and *closed*). The initialisation and expansion of the *bound* list are important because they are related to whether the Inequality (A.3) can be satisfied by Line 9 in Algorithm 9. For simplicity, the *bound* list is initialised by directly inserting all the unblocked search nodes in Zeta-SIPP, similar to how the *open* list is initialised in TO-AA-SIPP. However, the nodes in the *bound* list do not need to be examined by `lineOfSight`, thereby improving the performance of TO-AA-SIPP.

In `findNextClosedNode`, the SIPP-based collision detection with dynamic obstacles, namely `validateTransition`, will be executed to compute the real cost $g(bpp(n), n)$ from the best potential parent $bpp(n)$ to the current node n (Line 6). The path from $bpp(n)$ to n may be disturbed by dynamic obstacles, leading to $g(bpp(n), n) > h(bpp(n), n)$ in Eq. (A.2). Let the cost after the SIPP-based collision detection be

$$f_{sipp}(n) = g(bpp(n)) + g(bpp(n), n) + h(n) \tag{A.5}$$

Algorithm 5 Main Loop

while $\min_{n \in open} f(n) < \infty$ **do**
  $n \leftarrow$ `findNextClosedNode`$(open)$
  if $n \neq null$ **then**
    if $n = target$ **then**
      return `pathTo`(n)
    end if
    `invertedExpansion`$(n, open)$
  end if
  `forwardExpansion`$(open)$
end while
return $\varnothing$

Algorithm 6 `findNextClosedNode`$(open)$

1: $n \leftarrow \text{argmin}_{n \in open} f(n)$
2: remove n from $open$
3: **if** $bpp(n) \in potentialParents(n)$ **then**
4:   remove $bpp(n)$ from $potentialParents(n)$
5: **end if**
6: $g_{new} \leftarrow$ `validateTransition`$(bpp(n), n)$
7: **if** $g_{new} < g(n)$ **then**
8:   $g(n) \leftarrow g_{new}$
9:   $parent(n) \leftarrow bpp(n)$
10: **end if**
11: **if** `newBestPotentialParentExists`(n) **then**
12:   insert n into $open$
13:   **return** $null$
14: **end if**
15: **if** $g(n) < \infty$ and $g(n) + h(n) \leq \min_{n \in bound} f_{low}(n)$ and $g(n) + h(n) \leq \min_{n \in open} f(n)$ **then**
16:   insert n into $closed$
17:   **return** n
18: **else**
19:   insert n into $open$
20:   **return** $null$
21: **end if**

There maybe exists a "better" potential parent $pp(n)$ than $bpp(n)$ (Line 11, see Algorithm 7):

$$f(n) \leq f(n)' < f_{sipp}(n) \quad \exists pp(n) \in pps(n) \tag{A.6}$$

where $f(n)' = g(pp(n)) + h(pp(n), n) + h(n)$ and $pps(n)$ is the collection of potential parents of n. The best potential parent should be changed to $pp(n)$: $bpp(n) \leftarrow pp(n)$, as perhaps $f(n)' \leq f_{sipp}(n)' < f_{sipp}(n)$. In this case, the current node should be re-inserted

Algorithm 7 `newBestPotentialParentExists`(n)

$g_{low}(n) \leftarrow g(n)$, $bpp(n) \leftarrow parent(n)$
$f(n) \leftarrow g_{low}(n) + h(n)$
$isExisting \leftarrow false$
for each $n' \in potentialParents(n)$ **do**
    if $g(n') + h(n', n) < g_{low}(n)$ **then**
        $g_{low}(n) \leftarrow g(n') + h(n', n)$
        $bpp(n) \leftarrow n'$
        $f(n) \leftarrow g_{low}(n) + h(n)$
        $isExisting \leftarrow true$
    end if
end for
return $isExisting$

Algorithm 8 `addPotentialParent`(n, n')

insert n into $potentialParents(n')$
if $g(n) + h(n, n') < g_{low}(n')$ **then**
    $g_{low}(n') \leftarrow g(n) + h(n, n')$
    $bpp(n') \leftarrow n$
    $f(n') \leftarrow g_{low}(n') + h(n')$
    update n' in $open$
end if

into $open$ (Line 12) with a new cost $f(n) \leftarrow f(n)'$. Also, there maybe exists a "better" node n' than n (Line 15):

$$f(n) \leq f(n') < f_{sipp}(n) \quad \exists n' \in open \tag{A.7}$$

The current node n is no longer the most promising node as perhaps $f(n') \leq f_{sipp}(n') < f_{sipp}(n)$. In this case, the current node n should also be re-inserted into $open$ (Line 19) with a new cost $f(n) \leftarrow f_{sipp}(n)$. When the current node is not closed, `findNextClosedNode` will return $null$.

In Algorithm 8, n is a closed node while n' is an open node because only the closed node can be a potential parent of an open node. In Algorithm 9, the inverted expansion builds the connections between the current closed node and all the open nodes whereas the forward expansion inserts new nodes into $open$ based on the Inequality (A.3) and finds the connections between the new open nodes and all the closed nodes.

A.3.3. TO-AA-FoV-SIPP

The idea of TO-AA-FoV-SIPP is simple. At each step of TO-AA-SIPP, after a node is inserted into $closed$, shadowcasting will be executed to find visible open nodes from this closed node and subsequently this closed node will be added to their potential parent lists using Algorithm 8. This process is similar to `invertedExpansion` in Algorithm 9, but rather than `lineOfSight`, shadowcasting for computing the Field of View (FoV) is applied. In addition, the size of $open$ can be reduced using the Field of View. In TO-AA-FoV-SIPP,

Algorithm 9 Zeta-SIPP Expansion

```
function invertedExpansion(n, open)
    for each n′ ∈ open do
        if lineOfSight(n, n′) = true then
            addPotentialParent(n, n′)
        end if
    end for
end function

function forwardExpansion(open)
    while min_{n∈bound} f_{low}(n) ≤ min_{n∈open} f(n) do
        n ← argmin_{n∈bound} f_{low}(n)
        move n from bound to open
        invertedCheck(n)
    end while
end function

function invertedCheck(n)
    for each n′ ∈ closed do
        if lineOfSight(n′, n) = true then
            addPotentialParent(n′, n)
        end if
    end for
end function
```

only when a node is visible from one of the closed nodes can it be inserted into $open$. This is because if a node has no potential parent, the cost of this node is infinite. The inequality (A.4) can also be satisfied since $f(n) = \infty$, $\forall\, n \in S_{out}$ where $S_{out} = S \setminus (open \cup closed)$.

A.3.4. ZETA*-SIPP

The main pseudocode of Zeta*-SIPP is shown in Algorithm 10 and the major revision compared to Zeta-SIPP is marked in red. In general, the closed nodes may be the potential parents of the open nodes. Thus, the "light source" that needs to compute the Field of View should be a closed node, like TO-AA-FoV-SIPP. However, in Zeta*-SIPP, since the search range is bounded, *inverted scanning* is applied, which means that the open nodes are considered as "light sources" rather than the closed nodes. The "light sources" are distributed on the boundary of the search range and only illuminate the interior. At each step of the forward expansion, when a new node is moved from $bound$ to $open$, an inverted scanning is executed to compute the Field of View from this new open node and obtain the corresponding visible nodes inside the search range. In practice, a node n is treated as a wall in shadowcasting if its $f_{low}(n)$ is larger than the major axis length of the current elliptical boundary plus $\sqrt{2}$ grid length (grid buffer), and thus shadowcasting is bounded. Since the applied shadowcasting is symmetric, two nodes are mutually visible if one node can be seen from the other. These visible connections are recorded using

Algorithm 10 Zeta*-SIPP Expansion

1: **function** `invertedExpansion`(n)
2:     **for each** $n' \in children(n)$ **do**
3:         **if** $n' \notin closed$ **then**
4:             `addPotentialParent`(n, n')
5:         **end if**
6:     **end for**
7: **end function**

8: **function** `forwardExpansion`($open$)
9:     **while** $\min_{n \in bound} f_{low}(n) \leq \min_{n \in open} f(n)$ **do**
10:         $n \leftarrow \arg\min_{n \in bound} f_{low}(n)$
11:         move n from $bound$ to $open$
12:         `invertedScan`(n)
13:     **end while**
14: **end function**

15: **function** `invertedScan`(n)
16:     $N_{visible} \leftarrow$ `shadowcasting`(n) $\cap\, open$
17:     **for each** $n' \in N_{visible}$ **do**
18:         **if** $n' \in closed$ **then**
19:             `addPotentialParent`(n', n)
20:         **else**
21:             insert n' into $children(n)$
22:             insert n into $children(n')$
23:         **end if**
24:     **end for**
25: **end function**

Lines 21 and 22 of Algorithm 10. Therefore, when executing the inverted expansion in Zeta*-SIPP, there is no need to compute the Field of View for the closed nodes.

A.3.5. DATA STRUCTURE

In general, for SIPP-based planners, a node n can be represented by $(p, [t_1, t_2])$ where p is the location of the node n and $[t_1, t_2]$ is the safe interval. For the same location p, there may exist multiple nodes with different safe intervals. The node n can be regarded as a spatio-temporal point while the location p is only a spatial point. In graph-based path planning, p usually refers to a vertex or a grid. Therefore, the nodes in the proposed algorithms can be viewed as having two levels: the “node” level (space-time) and the “grid” level (space). Focusing solely on the “node” level may result in repeated visibility checks between two grids. To avoid this issue, the visibility checks can be conducted on the “grid” level and the results can then be stored in grids. If there is a need to check whether two nodes are mutually visible, the results can be generated or called from their corresponding grids.

A.4. THEORETICAL PROPERTIES

Zeta-SIPP, TO-AA-FoV-SIPP and Zeta*-SIPP have the same theoretical properties as TO-AA-SIPP. The focus is thus mainly on proving the properties affected by the $bound$ list in Zeta/Zeta*-SIPP.

Lemma 1. *The bound list always contains a node with the minimum f_{low}-value in the search space outside the open and closed lists $S \setminus (open \cup closed)$.*

Proof. Since the $bound$ list is initialised by inserting all the search nodes, according to Line 11 in Algorithm 10, $bound = S \setminus (open \cup closed)$. This concludes the proof. One can adapt different approaches to generate (or expand) the $bound$ list, but Lemma 1 must hold to ensure optimality. □

Lemma 2. *The node extracted from the open list at each step has the minimum f-value in the search space outside the closed list $S \setminus closed$.*

Proof. The while-loop in `forwardExpansion` (Algorithm 10) ensures $\min_{n \in open} f(n) < \min_{n \in bound} f_{low}(n)$ after expansion. Let $S_{out} = S \setminus (open \cup closed)$. Lemma 1 indicates $\min_{n \in bound} f_{low}(n) = \min_{n \in S_{out}} f_{low}(n)$. According to $f_{low}(n) \leq f(n)$, $\min_{n \in open} f(n) < \min_{n \in S_{out}} f(n)$. This concludes the proof. □

Theorem 1. *Zeta/Zeta*-SIPP is complete and optimal.*

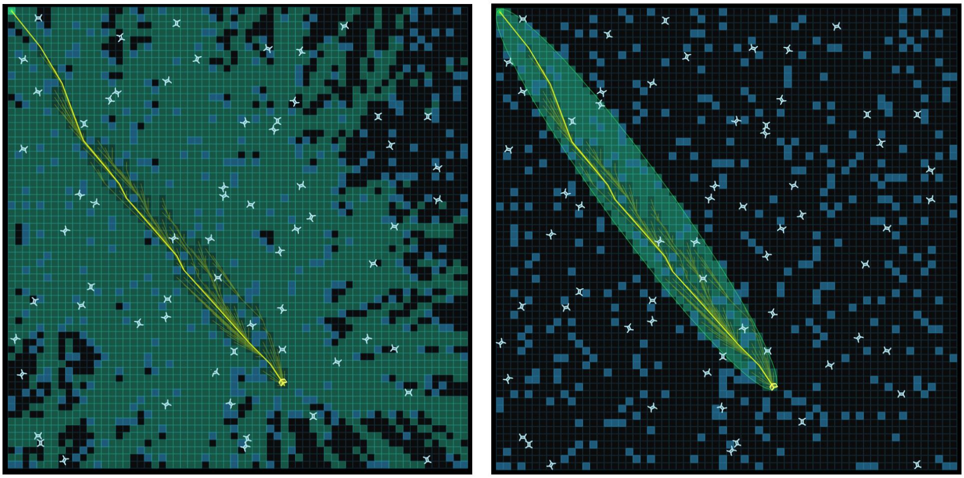
Proof. As TO-AA-SIPP has already proved to be complete and optimal, it is only needed to prove the main search procedure (Algorithm 6) of Zeta/Zeta*-SIPP is equivalent to that of TO-AA-SIPP. Let the open list of Zeta/Zeta*-SIPP be $open$ and the open list of TO-AA-SIPP be $open'$. In TO-AA-SIPP, since all search nodes are inserted into $open'$ at initialisation, $open' = S \setminus closed$. In Zeta/Zeta*-SIPP, Lemma 2 indicates $\min_{n \in open} f(n) = \min_{n \in S \setminus closed} f(n)$. Therefore, $\min_{n \in open} f(n) = \min_{n \in open'} f(n)$. This means that the node extracted from the open list at each step in Zeta/Zeta*-SIPP is identical to that of TO-AA-SIPP. However, Lemma 2 cannot guarantee this equation still holds after extraction. Therefore, it is required to add $g(n) + h(n) \leq \min_{n \in bound} f_{low}(n)$ in Line 15 of Algorithm 6. Then based on Lemma 1, the adjusted condition in Line 15 can prove to be equivalent to the original one in TO-AA-SIPP. □

A.5. EMPIRICAL ANALYSIS

TO-AA-FoV-SIPP, Zeta-SIPP and Zeta*-SIPP were implemented in a web-based pathfinding visualiser[1], as presented in Figure A.2. It is clear to see that Zeta/Zeta*-SIPP forms an elliptical boundary to limit the search range. The nodes outside the boundary have no contribution to finding the optimal path, and there is no need to insert them into $open$.

To assess the performance of the proposed algorithms, experiments were performed on three different benchmark maps [37]: *Random-64-64-10*, a 64 × 64 map with 10% of randomly blocked grids; *Warehouse-10-20-10-2-2*, a 170 × 84 map from a logistics domain; *Berlin_1_256*, a 256 × 256 real-world city map. These maps were chosen, as they

[1] URL: `http://dronectr.tudelft.nl/`, ID: zeta-sipp
Code implementation is available at `https://github.com/yiyuanzou/zeta-sipp`.



(a) TO-AA-FoV-SIPP (b) Zeta/Zeta*-SIPP

Figure A.2: Screenshots of TO-AA-FoV-SIPP and Zeta/Zeta*-SIPP

represent different types of environment and differ in size. For each map, 500 scenarios were generated by the following steps: 1) Chose 25 benchmark scenario sets (random) [37]. 2) For each scenario set, took the last 20 scenarios as tests and the top 32/64/96/128 scenarios as dynamic obstacles. 3) The trajectories of dynamic obstacles were generated successively by Zeta*-SIPP, which were collision-free and contained any-angle moves. The algorithms were implemented in JavaScript and the experiments were performed on Node.js v18.14.2 on a laptop with 2.30GHz Intel Core i7-11800H and 16 GB RAM.

Figure A.3 presents the mean algorithm runtime. On average, the proposed three algorithms all outperformed TO-AA-SIPP, and Zeta*-SIPP was the best among them. In the Random map, Zeta-SIPP has similar performance compared with Zeta*-SIPP. This is because when the map size is small, the advantage of Field of View is rather inconspicuous. In the Warehouse and Berlin maps, the performance difference between Zeta-SIPP and TO-AA-FoV-SIPP diminishes. When the dynamic obstacles are 32 and 64, TO-AA-FoV-SIPP is even faster than Zeta-SIPP. In the Berlin map, the mean runtime of TO-AA-SIPP surpasses 40-60 seconds. This indicates that the algorithm has relatively low efficiency, thereby limiting its applicability in real-world scenarios. Zeta*-SIPP substantially improves TO-AA-SIPP and the mean runtime is generally reduced by around 70%-90%.

Table A.1 shows the mean search nodes and scanned grids of the algorithms regardless of the number of dynamic obstacles. The search nodes indicate the nodes visited by the algorithms whereas the scanned grids mean the grids checked by Line of Sight or Field of View. Zeta/Zeta*-SIPP effectively reduces the search nodes of TO-AA-SIPP while maintaining the capability to find time-optimal paths. The search nodes are basically reduced by about 85%-95%. It is apparent that Field of View scans much fewer grids than Line of Sight, especially when the map is large. This also illustrates why TO-AA-FoV-SIPP performs better than Zeta-SIPP in the Warehouse and Berlin maps.

In Figure A.3 and Table A.1, means rather than medians are presented because me-

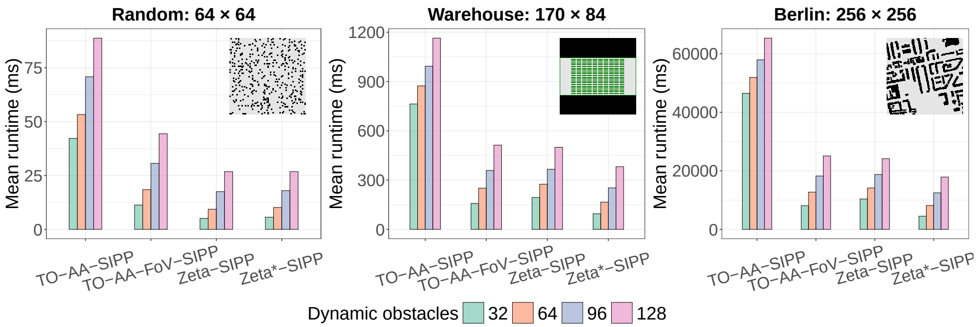


Figure A.3: Mean runtime of the algorithms.

Table A.1: Mean number of the search nodes and scanned grids

Maps	Search nodes (-SIPP)				Scanned grids (-SIPP)			
	TO-AA	TO-AA-FoV	Zeta	Zeta*	TO-AA	TO-AA-FoV	Zeta	Zeta*
Random	7.11×10^3	3.56×10^3	4.56×10^2	4.56×10^2	3.15×10^6	2.60×10^4	1.75×10^5	2.36×10^4
Warehouse	1.71×10^4	8.11×10^3	2.50×10^3	2.50×10^3	6.01×10^7	3.92×10^5	9.68×10^6	2.06×10^5
Berlin	6.20×10^4	2.53×10^4	8.53×10^3	8.53×10^3	4.31×10^9	1.46×10^7	7.07×10^8	6.94×10^6

dians may disregard some worst cases with extremely slow runtime. For example, when there are 32 dynamic obstacles in the Berlin map, relying solely on the median runtime may give the impression that Zeta*-SIPP is about 20 times faster than TO-AA-SIPP. However, in Figure A.3, the mean runtime of Zeta*-SIPP is around 10% of that of TO-AA-SIPP (only 10 times speedup). The improvements should be viewed with caution.

To further evaluate the improvements, Figure A.4 shows the frequency distributions of the percentage of reduced runtime in relation to the original TO-AA-SIPP runtime. The range of the x-axis is limited to $[0,1]$ since only the positive values represent improvements. It is worth noting that in 99.4% of scenarios, all the proposed algorithms outperformed TO-AA-SIPP. Hence, omitting the part less than 0 has little effect. The distributions illustrate the superiority of Zeta*-SIPP. In the majority (88.4%) of scenarios, Zeta*-SIPP reduces the runtime of TO-AA-SIPP by over 70%. Zeta-SIPP and Zeta*-SIPP both exhibit outstanding performance in the range $[0.9,1]$ compared with TO-AA-FoV-SIPP because their search ranges are limited by the elliptical region between the start and target. If the optimal path is almost a straight line, Zeta/Zeta*-SIPP may quickly find it and could be more than 10 times faster than TO-AA-SIPP.

A.6. CONCLUSION

Optimal any-angle path planning with dynamic obstacles remains relatively underexplored, with only a handful of algorithms making significant contributions to this area of research. TO-AA-SIPP is one of the important works in this field. However, the efficiency of this algorithm is an issue, posing difficulties for its applications. Therefore, two different directions to improve TO-AA-SIPP are proposed: 1) reduce useless search nodes by any-angle forward expansion (Zeta-SIPP), and 2) replace Line of Sight with Field of

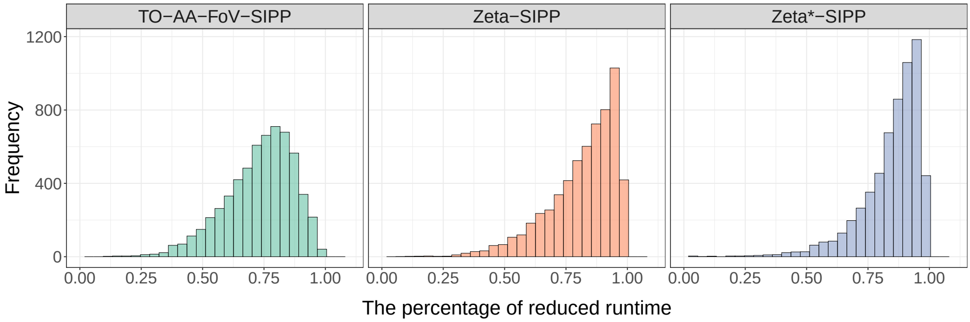


Figure A.4: The frequency distributions of the percentage of reduced runtime in relation to the original TO-AA-SIPP runtime.

View for visibility checks with static obstacles (TO-AA-FoV-SIPP). Combining these two ideas, Zeta*-SIPP is further developed. The initial experimental results of Zeta*-SIPP are promising, reducing the runtime of TO-AA-SIPP by around 70%-90% on average.

To enhance the applicability of the proposed algorithms in real-world scenarios, future work could consider incorporating domain-specific constraints. For instance, the constraints on turning angles could be involved to account for manoeuvrer restrictions of vehicles. In this case, the algorithms may be even faster as the search space is reduced. Furthermore, current algorithms only focus on 2D scenarios. In the future, 3D any-angle path planning with dynamic obstacles can also be explored (e.g., drone path planning).

B

QUESTIONNAIRES

This chapter presents two example questionnaires used in the experiments. The first questionnaire aims to assess participants' understanding of path-planning algorithms, with each question corresponding to a distinct transparency element (Chapter 3). Included as well are comments on the rationale behind the question design and observations gathered from the experiments. The second questionnaire is designed to explore the transparency needs and preferences of operators for supervising UTM systems (Chapter 4). It was developed and distributed through Qualtrics (a platform for online surveys). To conserve space, some repeated options are presented only once.

B.1. MEASURING UNDERSTANDING

1: The path-planning algorithm will always find the same path in the same situation or environment.
1) True.
2) False.
3) Unable to answer based on the available information.

Comments:
This question was designed based on the "path" element to test whether participants can correctly understand the algorithm's nature (deterministic vs. probabilistic). It might be easier to answer when the algorithm is probabilistic since providing a counterexample is simpler than proving it. However, several participants still failed to answer correctly for Informed RRT*, especially at lower levels of transparency, due to incorrect expectations about the algorithm and a lack of in-depth exploration of it.

2: The path-planning algorithm will always find the 'true shortest' path (if a path exists).
1) True.
2) False.
3) Unable to answer based on the available information.

Comments:
This question was designed based on the "path" element to test whether participants can correctly understand the algorithm's optimality. Due to the limited re-wiring radius and the restricted number of sampling points in the experiments, participants might easily recognise that the path found by Informed RRT* is not the true shortest. Compared to Informed RRT*, the path found by Theta* is deterministic and near-optimal, which might slightly increase the challenge for participants to answer this question correctly for Theta*.

3: What is the 'most accurate' statement regarding the constraint for the path found by the path-planning algorithm?
1) The path length cannot exceed a certain value.
2) The waypoints of the path cannot exceed the boundary of a fixed region.
3) The number of the waypoints cannot exceed a certain value.
4) The turning angle at the waypoint cannot exceed a certain value.
5) Unable to answer based on the available information.

Comments:
This question was designed based on the "search space" element to test whether participants can correctly understand the constraint via visualisation. The constraint implemented in the measurement phase was the first option: maximum path length. The difficulty of this question lies in the fact that if the first option is correct, then the second option is correct, and the third option is also not entirely wrong. This is because if the path length is limited, there must exist an elliptical region to constrain the entire search space (recall the "informed" procedure in Informed RRT*). As the number of waypoints increases, the path length will also increase according to the triangle inequality theorem.

4: What is the main advantage of the 'discretisation' method adopted by the path-planning algorithm?

1) Being able to represent the search space consistently and evenly regardless of the changes in obstacles.
2) Being able to build direct connections between the corners/vertices of obstacles before the algorithm begins its search.
3) Being able to represent the search space exactly using relatively few polygons, especially in open areas with few obstacles.
4) Being able to discretise the search space progressively and continuously while conducting the search, instead of using a fixed graph.
5) Unable to answer based on the available information.

Comments:

This question was designed based on the "graph" element to test whether participants can perceive the advantage of the discretisation method used in the algorithm. The first option is for regular grids, the second is for visibility graphs, the third is for navigation meshes, and the fourth is for sampling-based algorithms.

5: What is the correct statement regarding the 'search nodes' explored by the path-planning algorithm?

1) The search node is created based on the vertex (or centre) of a predefined graph.
2) The search node is randomly generated based on a predefined sampling strategy.
3) The search node is generated at the corner/vertex of an obstacle.
4) The search node represents a visible region from a certain location.
5) Unable to answer based on the available information.

Comments:

This question was designed based on the "explored node" element to test whether participants can correctly understand the concept of search nodes. The first option is for graph-based or grid-based algorithms, the second is for sampling-based algorithms, the third is for visibility-graph-based algorithms, and the fourth is for Anya and Polyanya. However, participants tended to select the fourth option, particularly after viewing the search trees. This tendency arose probably because a node is visible from its parent node (a certain location) and some participants failed to recognise that the node of the tested algorithm represents a point rather than a region. Additionally, the node of Theta* is presented as a grid cell instead of a point located at the grid centre, which may mislead participants about the algorithm.

6: What is the correct statement regarding the 'search tree' generated by the path-planning algorithm?

1) The search tree is predefined before the algorithm begins its search.
2) The search tree is randomly generated at the beginning.
3) The search tree is gradually expanded based on proximity and visibility.
4) The search tree does not change during the algorithm's execution.
5) Unable to answer based on the available information.

Comments:

This question was designed based on the "search tree" element to test whether participants can correctly understand the concept of search trees. The first option was to test whether participants can differentiate between predefined graphs and search trees. A search tree is the result of a search. The second option was designed as a trap, based on the fact that many algorithms in other fields generate a random initial solution as a starting point and then continuously optimise it. Also, the word "randomly" may mislead participants to associate this option with Informed RRT*. The fourth option was also designed as a trap. The deterministic algorithm (e.g., Theta*) consistently generates the same results and search trees, which might lead participants to believe that the fourth option is correct. Some participants were indeed misled.

7: What 'strategy' does the path-planning algorithm employ to attempt to find the shortest path?

1) Build direct visibility connections between the corners (or vertices) of obstacles by finding visible regions from explored locations.
2) Straighten the search tree by checking visibility from the current node's parent node to the current node's adjacent nodes.
3) Straighten the search tree by checking whether the current node could be a better parent of the nodes within its certain range.
4) Straighten the search tree by checking visibility between all explored nodes in the search space.
5) Unable to answer based on the available information.

Comments:

This question was designed based on the "search process" element to test whether participants can correctly understand the search strategy adopted by the algorithm. The first strategy is related to Anya and Polyanya, the second to Theta*, the third to RRT*-based algorithms and the fourth is borrowed from TO-AA-SIPP. This question is difficult to answer at lower levels of transparency.

8: What is the 'main disadvantage' of the path-planning algorithm?

1) Require significant preprocessing to discretise the search space before the algorithm begins its search, especially in complex environments (with many obstacles).
2) As the search space dimension increases (e.g., 3D, 4D, 5D, etc.), the time required to find a feasible path rises significantly (excluding the time for discretisation).
3) Poor performance in environments with narrow passages, occasionally being unable to pass through them.
4) Turning angles at waypoints are limited to specific integer increments (e.g., if the increment is 15°, the turning angle can only be one of the following: 15°, 30°, 45°, and so on.).
5) Unable to answer based on the available information.

Comments:

This question was designed to test participants' overall understanding, specifically whether they can identify the algorithm's disadvantage via visualisation. The first option pertains to generating irregular navigation meshes or visibility graphs, the second to grid-based path planning, the third to sampling-based algorithms, and the fourth is derived from the A* algorithm on grids (45-degree increments).

B

B.2. INVESTIGATING TRANSPARENCY NEEDS

B.2.1. INTRODUCTION

Thank you for participating in this survey! The survey aims to investigate your transparency needs for supervising Uncrewed Air Traffic Management (UTM) systems. First, you will be provided with a background to UTM. Then, we would like you to answer several questions that could help us identifying your needs for transparency. Please note that you can go back to the previous steps to view your responses. If you have any questions, feel free to ask the investigator(s).

B.2.2. BACKGROUND

Drone operations in urban areas are expected to significantly increase in number in the near future due to their large potential in medical deliveries, infrastructural inspections, area surveillance, etc. However, as indicated by several incidents (e.g, London Heathrow), the increase in drone traffic could disrupt and threaten crewed air traffic operating at low altitudes, like emergency helicopter flights and crewed aircraft during landing and take-off. Since 2017, ongoing efforts within the 'U-space project' initiated by the SESAR Joint Undertaking are developing procedures and technologies that aim to safely manage increased drone traffic in very low-level airspace (below 500 feet).

Recently, Rotterdam-The Hague Airport has opened up its lower airspace (below 500 feet AGL) to drone operations that should eventually be managed by U-space services once the system is fully operational. The figure below (see Figure 4.6a) shows representative drone missions superimposed on a map centred on Rotterdam-The Hague Airport. The blue line indicates potential medical delivery missions between hospitals of Rotterdam and The Hague. The Netherlands' expansive network of railway and highway infrastructure could also benefit from UAV-based inspection flights. Such inspection missions would need to closely follow railway (amber) and highway (pink) routes within the airport's control zone. Finally, the proximity of the airport to the Rotterdam harbour, which is one of the most important naval trade connections in Europe, could be problematic to any potential harbour inspection and surveillance flights by drones (purple). All of these missions would be performed almost entirely within the airport's control zone and in close proximity to the airport's runway.

Although the U-space vision expects that UTM relies on a set of highly automated services that require minimal human attention, 100% safe and reliable automation under all circumstances cannot be guaranteed. For that reason, some level of human supervision and/or intervention would still be required in case of abnormal system behaviour. From a safety perspective, Rotterdam-The Hague Airport has several geographical locations that pose increased risks of interactions with drones and crewed air traffic routes. For example, the locations where drone missions cross the extended runway centreline (e.g., the two ILS areas) that crewed aircraft follow on landing and departure. Additionally, many of the inspection routes coincide with published departure and arrival routes of aircraft operating under visual flight rules (VFR), providing the potential for separation losses and collisions between crewed and uncrewed aircraft. These locations with increased safety risks will need to be monitored by a human supervisor.

A prerequisite for adequate human supervision is that UTM is sufficiently 'transparent' to the human operator in order to understand what the system is currently doing, what it is planning to do next, and why. For example, UTM could automatically reroute drones in a specific way to solve a predicted separation problem with a crewed aircraft. Without some form of transparency, it could be difficult for human operators to understand why Solution A was selected by UTM instead of Solution B, and what underlying assumptions were considered in formulating that solution in the first place. For example, outdated and/or inaccurate information on environmental conditions (e.g., wind directions and magnitudes) could render the selected UTM solution infeasible and/or unsafe, requiring the human operator to intervene and override the UTM decision.

The goal of this survey is to gather your ideas and preferences on what information you would need to adequately supervise an automated UTM system and what type(s) of intervention(s) you prefer. This survey contains a set of questions that are centred around several hypothetical UTM scenarios at Rotterdam-The Hague Airport.

B.2.3. PERSONAL DETAILS

1: What is your highest level of education completed?
1) High school diploma or equivalent.
2) Bachelor's degree.
3) Master's degree.
4) Doctoral degree.
5) Other (please specify).

2: Please select the most applicable option regarding your air traffic control experience:
1) I am an active/retired tower controller.
2) I am an active/retired approach controller.
3) I am an active/retired area controller.
4) I am an air traffic control trainee.
5) I am an air traffic control researcher.
6) I am interested and familiar with air traffic control.
7) I know a little about air traffic control.
8) I have no idea what air traffic control is.
9) Other (please specify).

3: Please select the most applicable option regarding your drone operation experience: (Note: multiple answers can be selected)
1) I am a drone operator or pilot.
2) I am a drone developer/engineer or researcher.
3) I have some experience flying drones.
4) I have no experience with drones.
5) Other (please specify).

4: Please select the most applicable option regarding your experience in automation development:
1) Advanced: I have extensive experience in automation development and worked on several (complex) automation projects.
2) Intermediate: I have worked on a few automation projects and have a good grasp of automation tools and techniques.

B

3) Beginner: I have some basic understanding of automation development concepts and tools, but I haven't implemented any significant projects yet.
4) Novice: I have little to no experience in automation development.
5) Other (please specify).

B.2.4. TRANSPARENCY PREFERENCES

You need to rate the 20 transparency elements below on a 5-point Likert scale in different scenarios (Questions 5, 7, 9): Not at all, Slightly, Moderately, Very, and Extremely useful. After rating, group them accordingly (Question 11).

1) Aircraft routes: the routes of crewed and uncrewed aircraft.
2) Estimated state (e.g., remaining battery) and planned action (e.g., heading change) at each waypoint of the proposed path.
3) The underlying goals and intentions of the system to propose paths.
4) Predicted location of CPA if the drone follows the proposed path.
5) Predicted time to CPA if the drone follows the proposed path.
6) Predicted minimum separation (predicted separation at CPA) if the drone follows the proposed path.
7) Predicted probability of separation loss if the drone follows the proposed path.
8) Predicted location of separation loss if the drone would follow the old path.
9) Predicted start time of separation loss if the drone would follow the old path.
10) Predicted minimum separation (predicted separation at CPA) if the drone would follow the old path.
11) Predicted probability of separation loss if the drone would follow the old path.
12) Safe separation standards between crewed and uncrewed aircraft.
13) Drone manoeuvring space: the drone's flight range governed by battery power and environmental conditions such as wind.
14) Flight mission boundary (if applicable): certain drones, like surveillance drones, can only fly within a pre-approved mission area (e.g. railway inspections).
15) Wind field: wind speed and direction.
16) Search graph: a graph is usually how automation discretises a continuous space, and the generated path can only follow the edges of the graph.
17) Explored nodes: potential waypoints explored by the system.
18) Search tree: potential paths explored by the system.
19) The cost values of the explored potential paths given the system's goals and intentions.
20) Search process: a dynamic process that indicates how the search tree is composed and how the final path was found.
21) Other (please specify).

SCENARIO 1 (FIGURES 4.6B AND 4.6C)

A medical drone needs to pass through an area near the runway to deliver emergency supplies between two hospitals (from Rotterdam to Delft) as quickly and as safe as possible, but a crewed aircraft is about to land. The automated UTM conflict detection service has detected a potential separation problem (= conflict) between these two aircraft.

5: Sometimes the trajectory generated by the automated UTM conflict resolution service may not be what you expect (see Figure 4.6b). To help you with understanding **why**

and **how** the system generated a specific path, what information would you like to have? Please rate the following elements by your judged usefulness.

6: Please provide a brief explanation/motivation for how you rated the elements.

7: In some cases, the automated UTM service may not be able to find a feasible path (see Figure 4.6c). To understand why the system fails, what information would you like to see? Please rate the following elements by your judged usefulness.

8: Please provide a brief explanation/motivation for how you rated the elements.

SCENARIO 2 (FIGURE 4.6D)

When more drones need to cross the area covering the extended runway centreline, the conflict scenario may become more complex. Such increased complexity may have an impact on your transparency needs in the light of 'information overload'. Note that the UTM system in this investigation only deals with separation conflicts between crewed and uncrewed aircraft. The collision avoidance among drones is assumed to be achieved by 'sense and avoid' systems onboard drones. Therefore, the automated conflict detection & resolution services for drones only regard crewed aircraft as dynamic obstacles.

9: In Scenario 2, multiple drones are involved in a conflict, and you may **not have sufficient time** to check all information for understanding and supervision. In that case, what information would you like to see? Please rate the following elements by your judged usefulness. Note: The following information elements would be available for each drone.

10: Please provide a brief explanation/motivation for how you rated the elements.

GROUPING

Sometimes, you may wish to see several information elements at the same time to optimise your information processing efforts.

11: Please drag the information elements you think that belong together into the same box. Note that the number of groups is limited to 6 (some of these boxes can be empty) and the order of elements in each group (box) does not matter.

PROTOTYPES (FIGURES 4.3-4.5)

In the previous scenarios, we asked what information you would like to know. Next, you will see several suggestions on how and when transparency information could be presented to you. From here onwards, you cannot go back to the previous sections.

There are 14 visual prototypes in total. For each prototype, you need to rate its usefulness in both simple and complex scenarios using a 5-point Likert scale (Questions 12, 13): Not at all, Slightly, Moderately, Very, and Extremely useful. You also need to answer how you want to access the information element (Question 14).

12: Do you think the visualisation of this information element is useful in a **simple** scenario involving **only one drone** for understanding and supervising the UTM system?

13: Do you think the visualisation of this information element is useful in a **complex** scenario involving **multiple drones** for understanding and supervising the UTM system?

14: When would you like to present the visualisation of this information element?
1) Never presented.
2) Automatically activated for involved aircraft in a conflict event.
3) Globally activated by users on demand (e.g., buttons).
4) Click on or hover over aircraft (e.g., drones).
5) Click on or hover over a significant map location (e.g., a waypoint or grid).
6) Always presented.

B.2.5. INTERVENTION PREFERENCES

Since there are uncertainties that cannot be fully modelled into the UTM system, for example originating from wind prediction errors, uncertain drone kinematics & dynamics and drone navigation errors, the generated path cannot be 100% guaranteed to be conflict free at all times. This could make the path unsafe sometime later, requiring operator interventions in the UTM system. Additionally, the generated path may not align with operators' preferences on how to solve separation conflicts between uncrewed and crewed aircraft. Therefore, the UTM system should incorporate interaction methods that allow operators to intervene when required or desired.

You need to choose and rank the 6 intervention methods below in both simple and complex scenarios (Questions 15 and 16).

1) Give UTM instructions to directly change the drone altitude (note that the maximum altitude is 500ft).
2) Give UTM instructions to directly change the drone flight direction.
3) Give UTM instructions to directly change the drone route/waypoints.
4) Give UTM instructions to let the drone loiter in place.
5) Add intermediate waypoints to the drone: UTM will automatically plan the routes between these waypoints.
6) Activate geofences: UTM will automatically reroute drones to prevent it from flying through a shielded geographic location.
7) Other (please specify).

15: Which type of interaction with the UTM system do you prefer (for conflict resolution) in a **simple** scenario involving **only one drone** if the generated path is not satisfactory? Please drag a maximum of **three** of the elements below into the box and sort them by importance. The top most element represents the most useful element.

16: Which type of interaction with the UTM system do you prefer (for conflict resolution) in a **complex** scenario involving **multiple drones** if the generated path(s) is not satisfactory? Please drag a maximum of **three** of the elements below into the box and sort them by importance. The top most element represents the most useful element.

B.2.6. GENERAL OPINIONS

You need to answer Questions 17-19 using a 5-point Likert scale: Not at all, Slightly, Moderately, Very much, and Extremely. Questions 20-21 require free-text responses.

17: How important do you think transparency is in supervising the UTM system?

18: How much does transparency impact your level of acceptance and trust in the UTM system?

19: How much additional workload do you think transparency will bring to supervising the UTM system?

20: What other factors would affect your acceptance of a highly automated UTM system?

21: Do you have any additional final comment?

C

SIMULATORS

In this research, a unified code framework for simulators was developed based on JavaScript and OpenLayers. This framework was initially used to create a pathfinding visualiser for Chapters 2 and 3, which was later extended into a UTM simulator, called DroneCTR, for Chapters 4 and 5.

C

Figure C.1 presents the code framework for the developed simulators, comprising a database (e.g., flight plans and scenarios), a motion simulation module (e.g., aircraft kinematic models), an interactive interface (e.g., HTML, CSS, and JavaScript's "addEventListener"), and automation (e.g., models and algorithms). In this research, path planning is the core automation module, and the transparency component records, extracts, and displays information regarding the inner workings of automation.

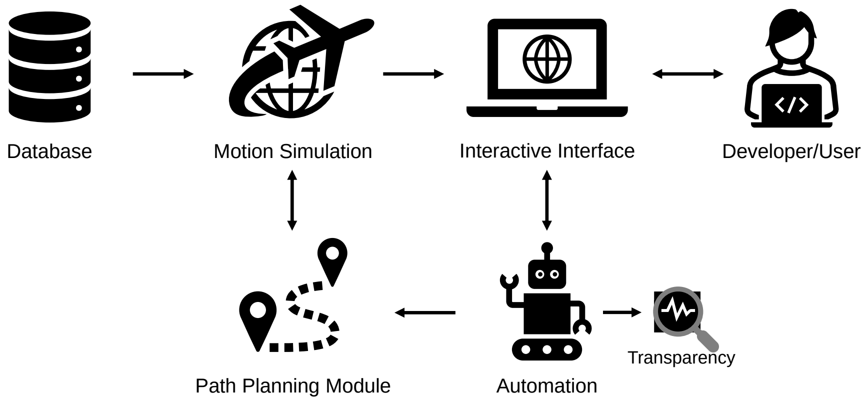


Figure C.1: Code framework of the developed simulators.

To achieve transparent path planning for UTM, various pathfinding visualisers were reviewed. However, these visualisers commonly showcase a limited selection of classic algorithms, such as Depth-First Search (DFS), Breadth-First Search (BFS), Greedy Best-First Search, Dijkstra's, and A*. Advanced algorithms typically are not included. For example, when reproducing Anya and Polyanya, only static images illustrating the algorithms' underlying concepts were available online. To the best of my knowledge, there were no interactive visualisers specifically designed for Anya and Polyanya, making it difficult for beginners to understand the algorithms' procedural details. Additionally, transparent path planning was intended not only for traditional algorithms like A*, but also for a wide range of advanced algorithms. Therefore, in Chapter 2, a general approach for path-planning visualisation was proposed and a novel pathfinding visualiser was developed, as shown in Figure 2.4.

Rather than only relying on the <canvas> HTML element, OpenLayers is utilised to draw features like grids and paths in the pathfinding visualiser. OpenLayers is an open-source JavaScript library for displaying interactive maps on web pages. Although OpenLayers is not necessary for pathfinding visualisers, it can facilitate their integration into map-based applications like ATM and UTM. Additionally, JavaScript was chosen over other programming languages because it allows easy deployment of the pathfinding visualiser on a website. This way, anyone accessing the link can interact with the algorithms and visualisations without any additional efforts, such as installing apps or having prior knowledge of specific programming languages.

In Chapters 4 and 5, the pathfinding visualiser was extended to a UTM simulator. As the focus was on UTM within Controlled Traffic Regions (CTR), the simulator was named

DroneCTR. The initial version of DroneCTR was developed based on Google Maps in [14]. However, Google Maps was not open-source, and thus the simulator was redesigned and migrated to OpenLayers, with the latest version shown in Figure 5.2.

All demos presented in this dissertation can be accessed through a unified link: `http://dronectr.tudelft.nl/` (see Figure C.2), using the following IDs:

1) **pathfinder**: pathfinding visualiser.
2) **zeta-sipp**: demo for Zeta*-SIPP.
3) **understanding**: demo for the experiment in Chapter 3.
4) **demo**: demo for UTM without transparency [63], basis for Chapter 4.
5) **transparency**: demo for the experiment in Chapter 5.

The entire project is hosted on GitLab, while some portions of the code are also publicly available on GitHub: `https://github.com/yiyuanzou/zeta-sipp`.

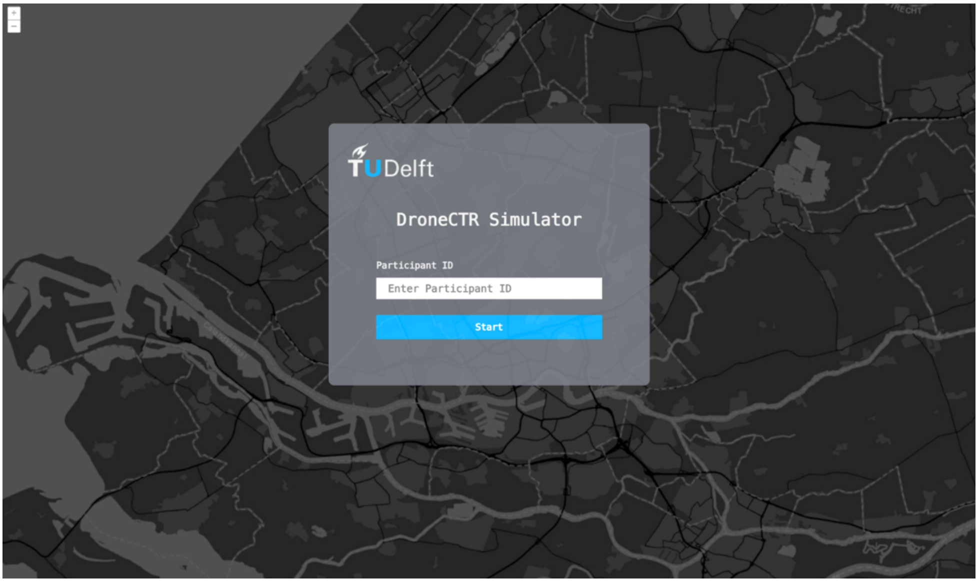


Figure C.2: Simulator login page.

References

[1] X. Guan, H. Shi, D. Xu, B. Zhang, J. Wei, and J. Chen, *The exploration and practice of low-altitude airspace flight service and traffic management in China,* Green Energy and Intelligent Transportation **3**, 100149 (2024).

[2] FAA, *FAA Aerospace Forecast Fiscal Years 2024-2044*, Tech. Rep. (Federal Aviation Administration, 2024).

[3] SESAR, *European drones outlook study: Unlocking the value for Europe*, Tech. Rep. (SESAR Joint Undertaking, 2016).

[4] EASA, *Drone Incident Management at Aerodromes Part 1: The challenge of unauthorised drones in the surroundings of aerodromes*, Tech. Rep. (European Union Aviation Safety Agency, 2021).

[5] P. H. Kopardekar, *Unmanned aerial system (UAS) traffic management (UTM): Enabling low-altitude airspace and UAS operations,* Tech. Rep. (National Aeronautics and Space Administration, 2014).

[6] P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. E. Robinson, *Unmanned aircraft system traffic management (UTM) concept of operations,* in *AIAA Aviation and Aeronautics Forum (Aviation 2016)*, ARC-E-DAA-TN32838 (2016).

[7] SESAR, *U-Space blueprint*, Tech. Rep. (SESAR Joint Undertaking, 2017).

[8] FAA, *UTM Concept of Operations Version 2.0*, Tech. Rep. (Federal Aviation Administration, 2022).

[9] SESAR, *U-space Concept of Operations fourth edition*, Tech. Rep. (SESAR Joint Undertaking, 2023).

[10] SESAR, *Development of the extended U-space concept of operations CORUS v5.0,* `https://www.sesarju.eu/projects/CORUS%20five` (2024), accessed on Oct. 1st, 2024.

[11] J. M. Hoekstra, J. Ellerbroek, E. Sunil, and J. Maas, *Geovectoring: Reducing traffic complexity to increase the capacity of UAV airspace,* in *International Conference for Research in Air Transportation, Barcelona, Spain* (2018) pp. 1–9.

[12] LVNL, *GoDrone,* `https://www.godrone.nl/` (2024), accessed on Oct. 3rd, 2024.

[13] D. van Aken, D. Janisch, and C. Borst, *Development and testing of a collaborative display for UAV traffic management and tower control,* in *Fourteenth USA/Europe Air Traffic Management Research and Development Seminar* (2021).

[14] D. Janisch, D. van Aken, and C. Borst, *Ecological collaborative interface for unmanned aerial vehicle traffic management and tower control,* Journal of Air Transportation **30**, 154–169 (2022).

[15] D. Janisch, P. Sánchez-Escalonilla, J. M. Cervero, A. Vidaller, and C. Borst, *Exploring tower control strategies for concurrent manned and unmanned aircraft management,* in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)* (2023) pp. 1–10.

[16] J. Teutsch and C. Petersen, *Dynamic airspace re-configuration for manned and unmanned operations in shared airspace,* in *2024 Integrated Communications, Navigation and Surveillance Conference (ICNS)* (2024) pp. 1–14.

[17] SESAR, *AURA (ATM U-space InteRfAce) Solution 2 Initial Concept Description*, Tech. Rep. (SESAR Joint Undertaking, 2021).

[18] F. Rajabiyazdi and G. A. Jamieson, *A review of transparency (seeing-into) models,* in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2020) pp. 302–308.

[19] A. Bhaskara, M. Skinner, and S. Loft, *Agent transparency: A review of current theory and evidence,* IEEE Transactions on Human-Machine Systems **50**, 215–224 (2020).

[20] G. A. Jamieson, G. Skraaning, and J. Joe, *The B737 MAX 8 accidents as operational experiences with automation transparency,* IEEE Transactions on Human-Machine Systems **52**, 794–797 (2022).

[21] J. B. Lyons, *Being transparent about transparency: A model for human-robot interaction,* in *2013 AAAI Spring Symposium Series* (2013) pp. 48–53.

[22] J. Y. Chen, K. Procci, M. Boyce, J. Wright, A. Garcia, and M. Barnes, *Situation awareness-based agent transparency,* US Army Research Laboratory , 1–29 (2014).

[23] J. E. Mercado, M. A. Rupp, J. Y. Chen, M. J. Barnes, D. Barber, and K. Procci, *Intelligent agent transparency in human–agent teaming for Multi-UxV management,* Human Factors **58**, 401–415 (2016).

[24] K. Stowers, N. Kasdaglis, M. A. Rupp, O. B. Newton, J. Y. Chen, and M. J. Barnes, *The impact of agent transparency on human performance,* IEEE Transactions on Human-Machine Systems **50**, 245–253 (2020).

[25] A. Bhaskara, L. Duong, J. Brooks, R. Li, R. McInerney, M. Skinner, H. Pongracic, and S. Loft, *Effect of automation transparency in the management of multiple unmanned vehicles,* Applied Ergonomics **90**, 103243 (2021).

[26] N. Diakopoulos and M. Koliska, *Algorithmic transparency in the news media,* Digital Journalism **5**, 809–828 (2017).

[27] S. Garfinkel, J. Matthews, S. S. Shapiro, and J. M. Smith, *Toward algorithmic transparency and accountability,* Communications of the ACM **60**, 5–5 (2017).

[28] Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan, *Dissecting racial bias in an algorithm used to manage the health of populations,* Science **366**, 447–453 (2019).

[29] T. Bitzer, M. Wiener, and W. A. Cram, *Algorithmic transparency: Concepts, antecedents, and consequences – a review and research framework,* Communications of the Association for Information Systems **52**, 293–331 (2023).

[30] A. Springer and S. Whittaker, *Progressive disclosure: When, why, and how do users want algorithmic transparency information?* ACM Transactions on Interactive Intelligent Systems (TiiS) **10**, 1–32 (2020).

[31] D. Gunning and D. Aha, *DARPA's explainable artificial intelligence (XAI) program,* AI Magazine **40**, 44–58 (2019).

[32] A. Adadi and M. Berrada, *Peeking inside the black-box: A survey on explainable artificial intelligence (XAI),* IEEE Access **6**, 52138–52160 (2018).

[33] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, *et al.*, *Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,* Information Fusion **58**, 82–115 (2020).

[34] E. Rader, K. Cotter, and J. Cho, *Explanations as mechanisms for supporting algorithmic transparency,* in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018) pp. 1–13.

[35] B. Goodman and S. Flaxman, *European union regulations on algorithmic decision-making and a 'right to explanation',* AI Magazine **38**, 50–57 (2017).

[36] SESAR, *AI in ATM: transparency, explainability, conformance, situation awareness and trust*, Tech. Rep. (SESAR Joint Undertaking, 2022).

[37] R. Stern, N. R. Sturtevant, A. Felner, S. Koenig, H. Ma, T. T. Walker, J. Li, D. Atzmon, L. Cohen, T. K. S. Kumar, E. Boyarski, and R. Bartak, *Multi-agent pathfinding: Definitions, variants, and benchmarks,* Symposium on Combinatorial Search (SoCS) , 151–158 (2019).

[38] W. Dai, B. Pang, and K. H. Low, *Conflict-free four-dimensional path planning for urban air mobility considering airspace occupancy,* Aerospace Science and Technology **119**, 107154 (2021).

[39] EASA, *Machine learning application approval,* `https://www.easa.europa.eu/en/research-projects/machine-learning-application-approval` (2024), accessed on Jul. 2nd, 2025.

[40] C. D. Hundhausen, S. A. Douglas, and J. T. Stasko, *A meta-study of algorithm visualization effectiveness,* Journal of Visual Languages & Computing **13**, 259–290 (2002).

[41] B. Aysolmaz and H. A. Reijers, *Animation as a dynamic visualization technique for improving process model comprehension,* Information & Management **58**, 103478 (2021).

[42] B. Tversky, J. B. Morrison, and M. Betrancourt, *Animation: can it facilitate?* International Iournal of Human-Computer Studies **57**, 247–262 (2002).

[43] N. R. Sturtevant, *Moving AI lab: Single agent search,* `https://www.movingai.com/SAS/`, accessed on Dec. 5th, 2023.

[44] X. Xu, *Pathfinding.js,* `https://github.com/qiao/PathFinding.js`, accessed on Dec. 5th, 2023.

[45] C. Mihailescu, *Pathfinding visualizer,* `https://github.com/clementmihailescu/Pathfinding-Visualizer`, accessed on Dec. 5th, 2023.

[46] ZJU-FAST-Lab, *C++ implementation and visualization of some sampling-based path planners,* `https://github.com/ZJU-FAST-Lab/sampling-based-path-finding`, accessed on Dec. 5th, 2023.

[47] A. Javaid, *Local planner visualization project,* `https://github.com/abdurj/Local-Planner-Visualization-Project`, accessed on Dec. 5th, 2023.

[48] A.-I. Toma, H.-Y. Hsueh, H. A. Jaafar, R. Murai, P. H. Kelly, and S. Saeedi, *Pathbench: A benchmarking platform for classical and learned path planning algorithms,* in *18th Conference on Robots and Vision (CRV)* (IEEE, 2021) pp. 79–86.

[49] E. W. Dijkstra, *A note on two problems in connexion with graphs,* Numer. Math. **1**, 269–271 (1959).

[50] P. E. Hart, N. J. Nilsson, and B. Raphael, *A formal basis for the heuristic determination of minimum cost paths,* IEEE Transactions on Systems Science and Cybernetics **4**, 100–107 (1968).

[51] J. Kuffner and S. LaValle, *RRT-connect: An efficient approach to single-query path planning,* in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065),* Vol. 2 (2000) pp. 995–1001.

[52] N. Sturtevant, *Benchmarks for grid-based pathfinding,* Transactions on Computational Intelligence and AI in Games **4**, 144 – 148 (2012).

[53] K. J. Vicente and J. Rasmussen, *Ecological interface design: Theoretical foundations,* IEEE Transactions on Systems, Man, and Cybernetics **22**, 589–606 (1992).

[54] M. Mulder, C. Borst, and M. M. van Paassen, *Improving operator situation awareness through ecological interfaces: Lessons from aviation,* in *1st International Conference on Computer-Human Interaction Research and Applications, CHIRA 2017* (Springer, 2019) pp. 20–44.

[55] G. Papadopoulos, A. Bastas, G. A. Vouros, I. Crook, N. Andrienko, G. Andrienko, and J. M. Cordero, *Deep reinforcement learning in service of air traffic controllers to resolve tactical conflicts,* Expert Systems with Applications **236**, 121234 (2024).

[56] R. Kilgore and M. Voshell, *Increasing the transparency of unmanned systems: Applications of ecological interface design,* in *Proceedings of the 6th International Conference on Virtual, Augmented and Mixed Reality. Applications of Virtual and Augmented Reality-Volume 8526* (2014) pp. 378–389.

[57] K. J. Vicente, *Cognitive Work Analysis: Towards Safe, Productive, and Healthy Computer-Based Work* (L. Erlbaum Associates Inc., USA, 1999).

[58] M. Dikmen, *A cognitive work analysis approach to explainable artificial intelligence in non-expert financial decision-making,* Phd thesis, University of Waterloo (2022).

[59] N. Pongsakornsathien, A. G. Gardi, R. Sabatini, and T. Kistan, *Evolutionary human-machine interactions for UAS traffic management,* in *AIAA Aviation 2021 Forum* (2021) p. 2337.

[60] C. Hurter, A. Degas, A. Guibert, N. Durand, A. Ferreira, N. Cavagnetto, M. R. Islam, S. Barua, M. U. Ahmed, S. Begum, *et al.*, *Usage of more transparent and explainable conflict resolution algorithm: Air traffic controller feedback,* Transportation Research Procedia **66**, 270–278 (2022).

[61] C. Westin, C. Borst, E. Kampen, T. M. Nunes, S. Boonsong, B. Hilburn, M. Cocchioni, and S. Bonelli, *Personalized and transparent AI support for ATC conflict detection and resolution: An empirical study,* in *Proceedings of the 12th SESAR Innovation Days* (2022) pp. 5–8.

[62] Y. Zou and C. Borst, *Investigating transparency needs for supervising unmanned air traffic management systems,* in *13th SESAR Innovation Days* (2023) pp. 1–9.

[63] D. Janisch, S. Wen, Y. Zou, and C. Borst, *Exploring the limits of uncrewed and crewed air traffic segregation by tower controllers,* in *14th SESAR Innovation Days* (2024) pp. 1–9.

[64] S. Karaman and E. Frazzoli, *Sampling-based algorithms for optimal motion planning,* The International Journal of Robotics Research **30**, 846–894 (2011).

[65] D. D. Harabor, A. Grastien, D. Öz, and V. Aksakalli, *Optimal any-angle pathfinding in practice,* Journal of Artificial Intelligence Research **56**, 89–118 (2016).

[66] K. Daniel, A. Nash, S. Koenig, and A. Felner, *Theta*: Any-angle path planning on grids,* Journal of Artificial Intelligence Research **39**, 533–579 (2010).

[67] N. Rivera, C. Hernández, N. Hormazábal, and J. A. Baier, *The 2^k neighborhoods for grid path planning,* Journal of Artificial Intelligence Research **67**, 81–113 (2020).

[68] R. Hechenberger, P. J. Stuckey, D. Harabor, P. Le Bodic, and M. A. Cheema, *Online computation of euclidean shortest paths in two dimensions,* in *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 30 (2020) pp. 134–142.

[69] B. Shen, M. A. Cheema, D. D. Harabor, and P. J. Stuckey, *Fast optimal and bounded suboptimal euclidean pathfinding,* Artificial Intelligence **302**, 103624 (2022).

[70] M. R. Endsley, *Supporting human-AI teams: Transparency, explainability, and situation awareness,* Computers in Human Behavior **140**, 107574 (2023).

[71] M. Brandao, A. Coles, and D. Magazzeni, *Explaining path plan optimality: Fast explanation methods for navigation meshes using full and incremental inverse optimization,* in *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 31 (2021) pp. 56–64.

[72] S. Almagor, J. Kottinger, and M. Lahijanian, *Temporal segmentation in multi agent path finding with applications to explainability,* Artificial Intelligence **330**, 104087 (2024).

[73] T. Chakraborti, S. Sreedharan, and S. Kambhampati, *The emerging landscape of explainable automated planning & decision making,* in *Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence* (2021) pp. 4803–4811.

[74] S. Sreedharan, T. Chakraborti, and S. Kambhampati, *Foundations of explanations as model reconciliation,* Artificial Intelligence **301**, 103558 (2021).

[75] M. Brandão and Y. Setiawan, *'why not this MAPF plan instead?' contrastive map-based explanations for optimal MAPF,* in *ICAPS 2022 Workshop on Explainable AI Planning* (2022).

[76] M. Brandao, G. Canal, S. Krivić, and D. Magazzeni, *Towards providing explanations for robot motion planning,* in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (2021) pp. 3927–3933.

[77] Q. Liu and M. Brandão, *Generating environment-based explanations of motion planner failure: Evolutionary and joint-optimization algorithms,* in *2024 IEEE International Conference on Robotics and Automation (ICRA)* (2024) pp. 15263–15269.

[78] A. Patel, *Red blob games,* `https://www.redblobgames.com/`, accessed on Dec. 5th, 2023.

[79] D. Misra, *Pathfinding visualizer threejs,* `https://github.com/dhruvmisra/Pathfinding-Visualizer-ThreeJS`, accessed on Dec. 5th, 2023.

[80] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, *Probabilistic roadmaps for path planning in high-dimensional configuration spaces,* IEEE Transactions on Robotics and Automation **12**, 566–580 (1996).

[81] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, *Batch informed trees (BIT*): Informed asymptotically optimal anytime search,* The International Journal of Robotics Research **39**, 543–567 (2020).

[82] M. L. Cui, D. D. Harabor, and A. Grastien, *Compromise-free pathfinding on a navigation mesh,* in *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (2017) pp. 496–502.

[83] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, *Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic,* in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2014) pp. 2997–3004.

[84] K. Yakovlev and A. Andreychuk, *Towards time-optimal any-angle path planning with dynamic obstacles,* in *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 31 (2021) pp. 405–414.

[85] Y. Zou and C. Borst, *Zeta*-SIPP: Improved time-optimal any-angle safe-interval path planning,* in *Proceedings of the 33rd International Joint Conference on Artificial Intelligence* (2024) pp. 6823–6830.

[86] S. R. Lawande, G. Jasmine, J. Anbarasi, and L. I. Izhar, *A systematic review and analysis of intelligence-based pathfinding algorithms in the field of video games,* Applied Sciences **12**, 5499 (2022).

[87] P. Yap, N. Burch, R. Holte, and J. Schaeffer, *Block A*: Database-driven search with applications in any-angle path-planning,* in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 25 (2011) pp. 120–125.

[88] M. Phillips and M. Likhachev, *SIPP: Safe interval path planning for dynamic environments,* in *2011 IEEE International Conference on Robotics and Automation* (2011) pp. 5628–5635.

[89] M. Penrose, *Random Geometric Graphs* (Oxford University Press, 2003).

[90] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, *Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs,* in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015) pp. 3067–3074.

[91] F. Xue and P. R. Kumar, *The number of neighbors needed for connectivity of wireless networks,* Wireless Networks **10**, 169–181 (2004).

[92] E. N. Gilbert, *Random plane networks,* Journal of the Society for Industrial and Applied Mathematics **9**, 533–543 (1961).

[93] M. T. Ribeiro, S. Singh, and C. Guestrin, *'why should i trust you?' explaining the predictions of any classifier,* in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016) pp. 1135–1144.

[94] S. M. Lundberg and S.-I. Lee, *A unified approach to interpreting model predictions,* Advances in Neural Information Processing Systems **30** (2017).

[95] Z. Juozapaitis, A. Koul, A. Fern, M. Erwig, and F. Doshi-Velez, *Explainable reinforcement learning via reward decomposition,* in *IJCAI/ECAI Workshop on Explainable Artificial Intelligence* (2019).

[96] C. Rudin, *Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,* Nature Machine Intelligence **1**, 206–215 (2019).

[97] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood, *Explainable AI for designers: A human-centered perspective on mixed-initiative co-creation,* in *2018 IEEE Conference on Computational Intelligence and Games (CIG)* (2018) pp. 1–8.

[98] J. Kottinger, S. Almagor, and M. Lahijanian, *MAPS-X: Explainable multi-robot motion planning via segmentation,* in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (2021) pp. 7994–8000.

[99] S. Almagor and M. Lahijanian, *Explainable multi agent path finding,* in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems* (2020) p. 34–42.

[100] K. Zheng, D. Harabor, and M. Wybrow, *Posthoc: A visualisation framework for understanding search,* in *ICAPS 2024 System's Demonstration Track* (2024) pp. 1–3.

[101] A. Tahirovic and M. Ferizbegovic, *Rapidly-exploring random vines (RRV) for motion planning in configuration spaces with narrow passages,* in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018) pp. 7055–7062.

[102] Z. Wu, Z. Meng, W. Zhao, and Z. Wu, *Fast-RRT: A RRT-based optimal path finding method,* Applied Sciences **11**, 11777 (2021).

[103] J. Cano, Y. Yang, B. Bodin, V. Nagarajan, and M. O'Boyle, *Automatic parameter tuning of motion planning algorithms,* in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2018) pp. 8103–8109.

[104] C. Kehoe, J. Stasko, and A. Taylor, *Rethinking the evaluation of algorithm animations as learning aids: An observational study,* International Journal of Human-Computer Studies **54**, 265–284 (2001).

[105] J. S. Park, R. Barber, A. Kirlik, and K. Karahalios, *A slow algorithm improves users' assessments of the algorithm's accuracy,* in *Proceedings of the ACM on Human-Computer Interaction*, Vol. 3 (2019) pp. 1–15.

[106] B. Bergström, *FOV using recursive shadowcasting,* `http://www.roguebasin.com/index.php?title=FOV_using_recursive_shadowcasting` (2001), accessed on Feb. 10th, 2023.

[107] M. M. van Paassen, C. Borst, J. Ellerbroek, M. Mulder, and J. M. Flach, *Ecological interface design for vehicle locomotion control,* IEEE Transactions on Human-Machine Systems **48**, 541–555 (2018).

[108] C. Borst, J. M. Flach, and J. Ellerbroek, *Beyond ecological interface design: Lessons from concerns and misconceptions,* IEEE Transactions on Human-Machine Systems **45**, 164–175 (2014).

[109] M. Mulder, C. Borst, and M. Van Paassen, *Designing for situation awareness aviation perspective,* in *2017 International Conference on Computer-Human Interaction Research and Applications, CHIRA 2017* (SciTePress, 2017) pp. 9–21.

[110] C. E. Billings, *Aviation automation: The search for a human-centered approach* (Lawrence Erlbaum Associates, Mahwah, NJ, 1997).

[111] M. Langer, D. Oster, T. Speith, H. Hermanns, L. Kästner, E. Schmidt, A. Sesing, and K. Baum, *What do we want from explainable artificial intelligence (XAI)? – a stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research,* Artificial Intelligence **296**, 103473 (2021).

[112] M. Bekier, B. R. C Molesworth, and A. M. Williamson, *Why air traffic controllers accept or refuse automated technology,* in *16th International Symposium on Aviation Psychology* (2011) p. 615.

[113] B. Hilburn, C. Westin, and C. Borst, *Will controllers accept a machine that thinks like they think? the role of strategic conformance in decision aiding automation,* Air Traffic Control Quarterly **22**, 115–136 (2014).

[114] S. Mohseni, N. Zarei, and E. D. Ragan, *A multidisciplinary survey and framework for design and evaluation of explainable AI systems,* ACM Transactions on Interactive Intelligent Systems (TiiS) **11**, 1–45 (2021).

[115] T. Miller, *Explanation in artificial intelligence: Insights from the social sciences,* Artificial intelligence **267**, 1–38 (2019).

[116] F. Doshi-Velez and B. Kim, *Towards a rigorous science of interpretable machine learning,* arXiv preprint (2017), arXiv:1702.08608.

[117] Z. C. Lipton, *The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery.* Queue **16**, 31–57 (2018).

[118] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, *From local explanations to global understanding with explainable AI for trees,* Nature Machine Intelligence **2**, 56–67 (2020).

[119] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, *et al.*, *Guidelines for human-AI interaction,* in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019) pp. 1–13.

[120] T. Lombrozo, *The structure and function of explanations,* Trends in Cognitive Sciences **10**, 464–470 (2006).

[121] R. R. Hoffman and G. Klein, *Explaining explanation part 1: Theoretical foundations,* IEEE Intelligent Systems **32**, 68–73 (2017).

[122] S. Sreedharan, A. O. Hernandez, A. P. Mishra, and S. Kambhampati, *Model-free model reconciliation,* in *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (2019) pp. 587–594.

[123] M. Brandao, G. Canal, S. Krivić, P. Luff, and A. Coles, *How experts explain motion planner output: A preliminary user-study to inform the design of explainable planners,* in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)* (IEEE, 2021) pp. 299–306.

[124] M. Brandao, M. Mansouri, A. Mohammed, P. Luff, and A. Coles, *Explainability in multi-agent path/motion planning: User-study-driven taxonomy and requirements,* in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems* (2022) pp. 172–180.

[125] S. Sreedharan, S. Srivastava, and S. Kambhampati, *Hierarchical expertise level modeling for user specific contrastive explanations,* in *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (2018) pp. 4829–4836.

[126] A. Tabrez, M. B. Luebbers, and B. Hayes, *A survey of mental modeling techniques in human–robot teaming,* Current Robotics Reports **1**, 259–267 (2020).

[127] T. L. Naps, G. Rößling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger, *et al.*, *Exploring the role of visualization and engagement in computer science education,* in *Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education* (2002) pp. 131–152.

[128] C. A. Shaffer, M. L. Cooper, A. J. D. Alon, M. Akbar, M. Stewart, S. Ponce, and S. H. Edwards, *Algorithm visualization: The state of the field,* ACM Transactions on Computing Education (TOCE) **10**, 1–22 (2010).

[129] M. R. Endsley, *Ironies of artificial intelligence,* Ergonomics **66**, 1656–1668 (2023).

[130] M. Eiband, H. Schneider, M. Bilandzic, J. Fazekas-Con, M. Haug, and H. Hussmann, *Bringing transparency design into practice,* in *Proceedings of the 23rd International Conference on Intelligent User Interfaces* (2018) pp. 211–223.

[131] D. Wang, Q. Yang, A. Abdul, and B. Y. Lim, *Designing theory-driven user-centric explainable AI,* in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019) pp. 1–15.

[132] A. S. Rao and M. P. George, *BDI agents: From theory to practice,* in *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)* (1995) pp. 312–319.

[133] J. D. Lee and K. A. See, *Trust in automation: Designing for appropriate reliance,* Human Factors **46**, 50–80 (2004).

[134] J. Y. Chen, S. G. Lakhmani, K. Stowers, A. R. Selkowitz, J. L. Wright, and M. Barnes, *Situation awareness-based agent transparency and human-autonomy teaming effectiveness,* Theoretical Issues in Ergonomics Science **19**, 259–282 (2018).

[135] M. R. Endsley, *Toward a theory of situation awareness in dynamic systems,* Human Factors **37**, 32–64 (1995).

[136] K. van de Merwe, S. Mallam, and S. Nazir, *Agent transparency, situation awareness, mental workload, and operator performance: A systematic literature review,* Human Factors **66**, 180–208 (2024).

[137] G. Vilone and L. Longo, *Notions of explainability and evaluation approaches for explainable artificial intelligence,* Information Fusion **76**, 89–106 (2021).

[138] C. Chen, S. Feng, A. Sharma, and C. Tan, *Machine explanations and human understanding,* Transactions on Machine Learning Research (2023), 10.1145/3593013.3593970.

[139] M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce, and H. P. Beck, *The role of trust in automation reliance,* International Journal of Human-Computer Studies **58**, 697–718 (2003).

[140] B. Y. Lim, A. K. Dey, and D. Avrahami, *Why and why not explanations improve the intelligibility of context-aware intelligent systems,* in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2009) pp. 2119–2128.

[141] F. Nothdurft, F. Richter, and W. Minker, *Probabilistic human-computer trust handling,* in *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)* (2014) pp. 51–59.

[142] H.-F. Cheng, R. Wang, Z. Zhang, F. O'connell, T. Gray, F. M. Harper, and H. Zhu, *Explaining decision-making algorithms through UI: Strategies to help non-expert stakeholders,* in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019) pp. 1–12.

[143] A. Bunt, M. Lount, and C. Lauzon, *Are explanations always important? a study of deployed, low-cost intelligent interactive systems,* in *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces* (2012) pp. 169–178.

[144] J. van der Waa, E. Nieuwburg, A. Cremers, and M. Neerincx, *Evaluating XAI: A comparison of rule-based and example-based explanations,* Artificial Intelligence **291**, 103404 (2021).

[145] T. Schmude, L. Koesten, T. Möller, and S. Tschiatschek, *On the impact of explanations on understanding of algorithmic decision-making,* in *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency* (2023) pp. 959–970.

[146] M. Nourani, S. Kabir, S. Mohseni, and E. D. Ragan, *The effects of meaningful and meaningless explanations on trust and perceived system accuracy in intelligent systems,* in *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 7 (2019) pp. 97–105.

[147] R. F. Kizilcec, *How much information? effects of transparency on trust in an algorithmic interface,* in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016) pp. 2390–2395.

[148] R. Kocielnik, S. Amershi, and P. N. Bennett, *Will you accept an imperfect AI? exploring designs for adjusting end-user expectations of AI systems,* in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019) pp. 1–14.

[149] M. Riveiro and S. Thill, *'that's (not) the output i expected!' on the role of end user expectations in creating explanations of AI systems,* Artificial Intelligence **298**, 103507 (2021).

[150] M. Dikmen and C. Burns, *The effects of domain knowledge on trust in explainable AI and task performance: A case of peer-to-peer lending,* International Journal of Human-Computer Studies **162**, 102792 (2022).

[151] J. S. Gurka and W. Citrin, *Testing effectiveness of algorithm animation,* in *Proceedings 1996 IEEE Symposium on Visual Languages* (1996) pp. 182–189.

[152] S. Grissom, M. F. McNally, and T. Naps, *Algorithm visualization in CS education: comparing levels of student engagement,* in *Proceedings of the 2003 ACM Symposium on Software Visualization* (2003) pp. 87–94.

[153] J. Rasmussen and K. J. Vicente, *Coping with human errors through system design: Implications for ecological interface design,* International Journal of Man-Machine Studies **31**, 517–534 (1989).

[154] J. Rasmussen, *The role of hierarchical knowledge representation in decisionmaking and system management,* IEEE Transactions on Systems, Man, and Cybernetics , 234–243 (1985).

[155] J. Urquiza-Fuentes and J. Á. Velázquez-Iturbide, *A survey of successful evaluations of program visualization and algorithm animation systems,* ACM Transactions on Computing Education (TOCE) **9**, 1–21 (2009).

[156] J. Sweller, *Cognitive load theory,* in *Psychology of Learning and Motivation*, Vol. 55 (Elsevier, 2011) pp. 37–76.

[157] B. McGuinness, *Quantitative analysis of situational awareness (quasa): Applying signal detection theory to true/false probes and self-ratings,* in *Command and Control Research and Technology Symposium June* (2004) pp. 15–17.

[158] P. Lopes, E. Silva, C. Braga, T. Oliveira, and L. Rosado, *XAI systems evaluation: A review of human and computer-centred methods,* Applied Sciences **12**, 9423 (2022).

[159] M. D. Svinicki, *A theoretical foundation for discovery learning.* Advances in Physiology Education **275**, S4 (1998).

[160] L. Alfieri, P. J. Brooks, N. J. Aldrich, and H. R. Tenenbaum, *Does discovery-based instruction enhance learning?* Journal of Educational Psychology **103**, 1 (2011).

[161] J. Michael, *Where's the evidence that active learning works?* Advances in Physiology Education (2006), 10.1152/advan.00053.2006.

[162] R. E. Mayer, *Should there be a three-strikes rule against pure discovery learning?* American Psychologist **59**, 14 (2004).

[163] R. Eisinga, T. Heskes, B. Pelzer, and M. Te Grotenhuis, *Exact p-values for pairwise comparison of friedman rank sums, with application to comparing classifiers,* BMC Bioinformatics **18**, 1–18 (2017).

[164] G. M. Sullivan and R. Feinn, *Using effect size — or why the p value is not enough,* Journal of Graduate Medical Education **4**, 279–282 (2012).

[165] C. I. Tsai, J. Klayman, and R. Hastie, *Effects of amount of information on judgment accuracy and confidence,* Organizational Behavior and Human Decision Processes **107**, 97–105 (2008).

[166] M. Daun, J. Brings, P. A. Obe, and V. Stenkova, *Reliability of self-rated experience and confidence as predictors for students' performance in software engineering: Results from multiple controlled experiments on model comprehension with graduate and undergraduate students,* Empirical Software Engineering **26**, 80 (2021).

[167] K. Doherty and G. Doherty, *Engagement in HCI: conception, theory and measurement,* ACM Computing Surveys (CSUR) **51**, 1–39 (2018).

[168] M. R. Endsley and E. O. Kiris, *The out-of-the-loop performance problem and level of control in automation,* Human Factors **37**, 381–394 (1995).

[169] L. Bainbridge, *Ironies of automation,* Automatica **19**, 775–779 (1983).

[170] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viegas, and J. Wilson, *The what-if tool: Interactive probing of machine learning models,* IEEE Transactions on Visualization & Computer Graphics **26**, 56–65 (2020).

[171] G. L. Calhoun, H. A. Ruff, K. J. Behymer, and E. M. Frost, *Human-autonomy teaming interface design considerations for multi-unmanned vehicle control,* Theoretical Issues in Ergonomics Science **19**, 321–352 (2018).

[172] A. Arguel, L. Lockyer, O. V. Lipp, J. M. Lodge, and G. Kennedy, *Inside out: detecting learners' confusion to improve interactive digital learning environments,* Journal of Educational Computing Research **55**, 526–551 (2017).

[173] M. F. B. Mohamed Salleh and K. H. Low, *Concept of operations (ConOps) for traffic management of unmanned aircraft systems (TM-UAS) in urban environment,* in *AIAA Information Systems-AIAA Infotech @ Aerospace* (2017).

[174] FAA, *Unmanned Aircraft Systems Traffic Management (UTM) Implementation Plan,* Tech. Rep. (Federal Aviation Administration, 2023).

[175] T. J. Lieb and T. F. Sievers, *A comparative analysis of the European U-space framework and the US-American UAS traffic management (UTM) system,* in *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)* (IEEE, 2024) pp. 1–9.

[176] J. Y. Chen, F. O. Flemisch, J. B. Lyons, and M. A. Neerincx, *Guest editorial: Agent and system transparency,* IEEE Transactions on Human-Machine Systems **50**, 189–193 (2020).

[177] G. Schwoch, T. J. Lieb, M. Shamim, and G. Vanhandenhove, *Interaction between ATM and UAS operators in U-space operations and potential automation benefits,* in *2024 Integrated Communications, Navigation and Surveillance Conference (ICNS)* (2024) pp. 1–9.

[178] W. Zhang, D. Feltner, D. Kaber, and J. Shirley, *Utility of functional transparency and usability in UAV supervisory control interface design,* International Journal of Social Robotics **13**, 1761–1776 (2021).

[179] CORDIS, *Bringing intelligent and trustworthy automation to Europe's aviation sector*, Tech. Rep. (European Commission, 2022).

[180] A. P. Saraf, K. Chan, M. Popish, J. Browder, and J. Schade, *Explainable artificial intelligence for aviation safety applications,* in *AIAA AVIATION 2020 FORUM* (2020) p. 2881.

[181] C. S. Hernandez, S. Ayo, and D. Panagiotakopoulos, *An explainable artificial intelligence (XAI) framework for improving trust in automated ATM tools,* in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)* (2021) pp. 1–10.

[182] K. Wang, W. Hou, L. Hong, and J. Guo, *Smart transparency: A user-centered approach to improving human–machine interaction in high-risk supervisory control tasks,* Electronics **14**, 420 (2025).

[183] Y. Xie, N. Pongsakornsathien, A. Gardi, and R. Sabatini, *Explanation of machine-learning solutions in air-traffic management,* Aerospace **8**, 224 (2021).

[184] K. J. Vicente and J. Rasmussen, *The ecology of human-machine systems ii: Mediating 'direct perception' in complex work domains,* Ecological Psychology **2**, 207–249 (1990).

[185] C. F. Michaels and C. Carello, *Direct perception* (Prentice-Hall Englewood Cliffs, NJ, 1981).

[186] C. Fuchs, C. Borst, G. C. de Croon, M. Van Paassen, and M. Mulder, *An ecological approach to the supervisory control of UAV swarms,* International Journal of Micro Air Vehicles **6**, 211–229 (2014).

[187] C. Borst, V. A. Bijsterbosch, M. Van Paassen, and M. Mulder, *Ecological interface design: Supporting fault diagnosis of automated advice in a supervisory air traffic control task,* Cognition, Technology & Work **19**, 545–560 (2017).

[188] C. Borst, R. M. Visser, M. M. van Paassen, and M. Mulder, *Exploring short-term training effects of ecological interfaces: A case study in air traffic control,* IEEE Transactions on Human-Machine Systems **49**, 623–632 (2019).

[189] G. A. M. Velasco, C. Borst, M. M. van Paassen, and M. Mulder, *Solution space decision support for reducing controller workload in route merging task,* Journal of Aircraft **58**, 125–137 (2021).

[190] J. Flach, *Supporting productive thinking: The semiotic context for cognitive systems engineering (cse),* Applied Ergonomics **59**, 612–624 (2017).

[191] L. J. Planke, Y. Lim, A. Gardi, R. Sabatini, T. Kistan, and N. Ezer, *A cyber-physical-human system for one-to-many UAS operations: Cognitive load analysis,* Sensors **20**, 5467 (2020).

[192] W. Zhang, Y. Liu, and D. B. Kaber, *Effect of interface design on cognitive workload in unmanned aerial vehicle control,* International Journal of Human-Computer Studies **189**, 103287 (2024).

[193] EASA, *Artificial Intelligence Roadmap 2.0: A human-centric approach to AI in aviation*, Tech. Rep. (European Union Aviation Safety Agency, 2023).

[194] B. J. van Marwijk, C. Borst, M. Mulder, M. Mulder, and M. M. van Paassen, *Supporting 4D trajectory revisions on the flight deck: Design of a human–machine interface,* The International Journal of Aviation Psychology **21**, 35–61 (2011).

[195] R. Klomp, R. Riegman, C. Borst, M. Mulder, and M. Van Paassen, *Solution space concept: Human-machine interface for 4D trajectory management,* in *Proceedings of the 13th USA/Europe Air Traffic Management Research and Development Seminar (ATM2019), Vienna, Austria* (2019) pp. 17–21.

[196] J. Cha, M. Barnes, and J. Chen, *Visualization techniques for transparent human-agent interface designs*, Tech. Rep. (US Combat Capabilities Development Command Army Research Laboratory Aberdeen Proving Ground, 2019).

[197] A. A. Alharbi, I. Petrunin, and D. Panagiotakopoulos, *Modeling and characterization of traffic flow patterns and identification of airspace density for UTM application,* IEEE Access **10**, 130110–130134 (2022).

[198] N. K. Koerkamp, C. Borst, M. Mulder, and M. Van Paassen, *Supporting humans in solving multi-UAV dynamic vehicle routing problems,* IFAC-PapersOnLine **52**, 359–364 (2019).

[199] P. Pons and M. Latapy, *Computing communities in large networks using random walks,* in *Computer and Information Sciences-ISCIS 2005: 20th International Symposium, Istanbul, Turkey* (Springer, 2005) pp. 284–293.

[200] B. M. King, P. J. Rosopa, and E. W. Minium, *Statistical reasoning in the behavioral sciences* (John Wiley & Sons, 2018).

[201] C. Westin, C. Borst, and B. Hilburn, *Strategic conformance: Overcoming acceptance issues of decision aiding automation?* IEEE Transactions on Human-Machine Systems **46**, 41–52 (2015).

[202] L. Simon, P. Rauffet, and C. Guerin, *Effects of agent transparency and situation criticality upon human-autonomy trust and risk perception in decision-making,* Cognition, Technology & Work , 1–18 (2024).

[203] J. J. Van Merriënboer and L. Kester, *The four-component instructional design model: Multimedia principles in environments for complex learning,* The Cambridge Handbook of Multimedia Learning , 71–93 (2005).

[204] J. J. Van Merriënboer, *The four-component instructional design model,* Open Education Research **26**, 35–43 (2019).

[205] D. Groot, M. Ribeiro, J. Ellerbroek, and J. Hoekstra, *Policy analysis of safe vertical manoeuvring using reinforcement learning: Identifying when to act and when to stay idle.* in *13th SESAR Innovation Days* (2023).

[206] SESAR, *Demonstrating everyday benefits of U-space*, Tech. Rep. (SESAR 3 Joint Undertaking, 2022).

[207] N. Patrinopoulou, I. Daramouskas, V. Lappas, V. Kostopoulos, A. M. Veytia, C. A. Badea, J. Ellerbroek, J. Hoekstra, V. de Vries, J. van Ham, *et al.*, *Metropolis II: Investigating the future shape of air traffic control in highly dense urban airspace,* in *2022 30th Mediterranean Conference on Control and Automation (MED)* (IEEE, 2022) pp. 649–655.

[208] M. Gianfelice, H. Aboshosha, and T. Ghazal, *Real-time wind predictions for safe drone flights in Toronto,* Results in Engineering **15**, 100534 (2022).

[209] SESAR, *ENSURE- ATM-Uspace interface and airspace reconfiguration service,* `https://www.sesarju.eu/projects/ensure` (2023), accessed on Oct. 1st, 2024.

[210] SESAR, *MUSE- measuring U-Space social and environmental impact,* `https://www.sesarju.eu/projects/MUSE` (2023), accessed on Oct. 1st, 2024.

[211] K. van de Merwe, S. Mallam, S. Nazir, and Ø. Engelhardtsen, *Supporting human supervision in autonomous collision avoidance through agent transparency,* Safety Science **169**, 106329 (2024).

[212] N. Pongsakornsathien, A. Gardi, R. Sabatini, T. Kistan, and N. Ezer, *Human-machine interactions in very-low-level UAS operations and traffic management,* in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)* (2020) pp. 1–8.

[213] Y. Lim, N. Pongsakornsathien, A. Gardi, R. Sabatini, T. Kistan, N. Ezer, and D. J. Bursch, *Adaptive human-robot interactions for multiple unmanned aerial vehicles,* Robotics **10**, 12 (2021).

[214] Y. Zou and C. Borst, *Algorithmic transparency in path planning: A visual approach to enhancing human understanding,* International Journal of Human-Computer Studies **203**, 103573 (2025).

[215] M. Harbers, K. Van Den Bosch, and J.-J. Meyer, *Design and evaluation of explainable BDI agents,* in *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Vol. 2 (IEEE, 2010) pp. 125–132.

[216] H. Kopecka, J. Such, and M. Luck, *The role of perception, acceptance, and cognition in the usefulness of robot explanations,* in *Proceedings of the 33rd International Joint Conference on Artificial Intelligence* (2024) pp. 7868–7876.

[217] M. Harbers, D. Broekens, K. van den Bosch, and J. Meyer, *Guidelines for developing explainable cognitive models,* in *10th International Conference on Cognitive Modeling (ICCM 2010)* (ICCM, 2010) pp. 85–90.

[218] M. M. De Graaf and B. F. Malle, *People's explanations of robot behavior subtly reveal mental state inferences,* in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (2019) pp. 239–248.

[219] S. Kaplan, H. Uusitalo, and L. Lensu, *A unified and practical user-centric framework for explainable artificial intelligence,* Knowledge-Based Systems **283**, 111107 (2024).

[220] C. M. Burns, G. Skraaning Jr, G. A. Jamieson, N. Lau, J. Kwok, R. Welch, and G. Andresen, *Evaluation of ecological interface design for nuclear process control: Situation awareness effects,* Human Factors **50**, 663–679 (2008).

[221] J. Rasmussen, *Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models,* IEEE Transactions on Systems, Man, and Cybernetics , 257–266 (1983).

[222] L. Simon, C. Guérin, P. Rauffet, and J.-P. Diguet, *Integrating transparency to ecological interface design,* in *Proceedings of the 32nd European Safety and Reliability Conference (ESREL 2022)* (2022) pp. 3301–3308.

[223] X. J. Yang, V. V. Unhelkar, K. Li, and J. A. Shah, *Evaluating effects of user experience and system transparency on trust in automation,* in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction* (2017) pp. 408–416.

[224] D. Shin, *The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable AI,* International Journal of Human-Computer Studies **146**, 102551 (2021).

[225] A. Bussone, S. Stumpf, and D. O'Sullivan, *The role of explanations on trust and reliance in clinical decision support systems,* in *2015 International Conference on Healthcare Informatics* (IEEE, 2015) pp. 160–169.

[226] L. Perlmutter, E. Kernfeld, and M. Cakmak, *Situated language understanding with human-like and visualization-based transparency.* in *Robotics: Science and Systems,* Vol. 12 (2016) pp. 40–50.

[227] I. Gegoff, M. Tatasciore, V. Bowden, J. McCarley, and S. Loft, *Transparent automated advice to mitigate the impact of variation in automation reliability,* Human Factors **66**, 2008–2024 (2024).

[228] M. Tatasciore and S. Loft, *Can increased automation transparency mitigate the effects of time pressure on automation use?* Applied Ergonomics **114**, 104142 (2024).

[229] K. van de Merwe, S. Mallam, S. Nazir, and Ø. Engelhardtsen, *The influence of agent transparency and complexity on situation awareness, mental workload, and task performance,* Journal of Cognitive Engineering and Decision Making **18**, 156–184 (2024).

[230] S. Loft, A. Bhaskara, B. A. Lock, M. Skinner, J. Brooks, R. Li, and J. Bell, *The impact of transparency and decision risk on human–automation teaming outcomes,* Human Factors **65**, 846–861 (2023).

[231] G. Frontera, J. Cuadrado, E. Bartolomé, and C. Querejeta, *Skyway simulator: An integrated ATM/UTM simulator for autonomous operations,* in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)* (2021) pp. 1–9.

[232] S. Yoon, D. Shin, Y. Choi, and K. Park, *Development of a flexible and expandable UTM simulator based on open sources and platforms,* Aerospace **8**, 133 (2021).

[233] V. Lappas, G. Zoumponos, V. Kostopoulos, H. I. Lee, H.-S. Shin, A. Tsourdos, M. Tantardini, D. Shomko, J. Munoz, E. Amoratis, *et al.*, *Eurodrone, a European unmanned traffic management testbed for U-space,* Drones **6**, 53 (2022).

[234] F. Dehais, V. Peysakhovich, S. Scannella, J. Fongue, and T. Gateau, *"automation surprise" in aviation: real-time solutions,* in *Proceedings of the 33rd annual ACM conference on Human Factors in Computing Systems* (2015) pp. 2525–2534.

[235] G. de Rooij, A. B. Tisza, and C. Borst, *Flight-based control allocation: Towards human–autonomy teaming in air traffic control,* Aerospace **11**, 919 (2024).

[236] P. S. Farrugia and A. Micallef, *Vectorial statistics for the standard deviation of wind direction,* Meteorology and Atmospheric Physics **129**, 495–506 (2017).

[237] S. G. Hart, *NASA-task load index (NASA-TLX); 20 years later,* Proceedings of the Human Factors and Ergonomics Society Annual Meeting **50**, 904–908 (2006).

[238] B. Donmez, M. L. Cummings, H. D. Graham, and A. S. Brzezinski, *Modified cooper harper scales for assessing unmanned vehicle displays,* in *Proceedings of the 10th Performance Metrics for Intelligent Systems Workshop* (2010) pp. 235–242.

[239] K. Lee, K. Kerns, R. Bone, and M. Nickelson, *Development and validation of the controller acceptance rating scale (cars): Results of empirical research,* in *Proceedings of the 4th USA/Europe Air Traffic Management R&D Seminar* (2001).

[240] C. Hamamura, *radarboxplot: Implementation of the Radar-Boxplot,* (2021).

[241] B. Shneiderman, *Direct manipulation: A step beyond programming languages,* Computer **16**, 57–69 (1983).

[242] R. Klomp, C. Borst, R. van Paassen, and M. Mulder, *Expertise level, control strategies, and robustness in future air traffic control decision aiding,* IEEE Transactions on Human-Machine Systems **46**, 255–266 (2015).

[243] Y. K. Hwang, N. Ahuja, *et al.*, *A potential field approach to path planning.* IEEE transactions on robotics and automation **8**, 23–32 (1992).

[244] A. Tamar, Y. Wu, G. Thomas, S. Levine, and P. Abbeel, *Value iteration networks,* Advances in neural information processing systems **29** (2016).

[245] J. Lundberg and B. J. Johansson, *A framework for describing interaction between human operators and autonomous, automated, and manual control systems,* Cognition, Technology & Work **23**, 381–401 (2021).

[246] J. Ren, X. Huang, and R. N. Huang, *Efficient deep reinforcement learning for optimal path planning,* Electronics **11**, 3628 (2022).

[247] D. Kirilenko, A. Andreychuk, A. Panov, and K. Yakovlev, *Transpath: Learning heuristics for grid-based pathfinding via transformers,* in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37 (2023) pp. 12436–12443.

[248] J. Wang, W. Chi, C. Li, C. Wang, and M. Q.-H. Meng, *Neural RRT*: Learning-based optimal path planning,* IEEE Transactions on Automation Science and Engineering **17**, 1748–1758 (2020).

[249] M. J. Lazaro, Y. Kang, M. H. Yun, and S. Kim, *The effects of visual complexity and decluttering methods on visual search and target detection in cockpit displays,* International Journal of Human–Computer Interaction **37**, 588–600 (2021).

[250] V. Vagal, K. Markantonakis, and C. Shepherd, *A new approach to complex dynamic geofencing for unmanned aerial vehicles,* in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)* (IEEE, 2021) pp. 1–7.

[251] T. Uras and S. Koenig, *An empirical comparison of any-angle path-planning algorithms,* in *Proceedings of the International Symposium on Combinatorial Search*, Vol. 6 (2015) pp. 206–210.

[252] D. Silver, *Cooperative pathfinding,* in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 1 (2005) pp. 117–122.

[253] K. Yakovlev and A. Andreychuk, *Any-angle pathfinding for multiple agents based on SIPP algorithm,* in *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 27 (2017) pp. 586–594.

[254] A. Nash, S. Koenig, and C. Tovey, *Lazy Theta*: Any-angle path planning and path length analysis in 3D,* in *Proceedings of the AAAI Conference on Artificial Intelligence,* Vol. 24 (2010) pp. 147–154.

[255] J. E. Bresenham, *Algorithm for computer control of a digital plotter,* IBM Systems Journal **4**, 25–30 (1965).

Abbreviations

4C/ID	Four-Component Instructional Design
AA-SIPP	Any-Angle SIPP
AGL	Above Ground Level
AH	Abstraction Hierarchy
AI	Artificial Intelligence
ATC	Air Traffic Control
ATCo	Air Traffic Controller
ATM	Air Traffic Management
AURA	ATM U-space Interface
BDI	Belief-Desire-Intention framework
BIT	Batch Informed Tree
CD&R	Conflict Detection & Resolution
ConOps	Concept of Operations
CPA	Closest Point of Approach
CS	Computer Science
CTR	Controlled Traffic Region
CWA	Cognitive Work Analysis
DAR	Dynamic Airspace Reconfiguration
drDQN	decomposed reward Deep Q-Networks
EASA	European Union Aviation Safety Agency
EID	Ecological Interface Design
EU	European Union
FAA	Federal Aviation Administration
FCFS	First-Come-First-Served
FoV	Field of View
GDPR	General Data Protection Regulation
HCI	Human-Computer Interaction
HREC	Human Research Ethics Committee
IFR	Instrument Flight Rules
JCF	Joint Control Framework
LIME	Local Interpretable Model-agnostic Explanations
LoS	Line of Sight
LPVP	Local Planner Visualization Project
LVNL	Luchtverkeersleiding Nederland
MAPF	Multi-Agent Path Finding
NASA	National Aeronautics and Space Administration
NN	Neural Networks
PRM	Probabilistic Roadmap

RGG	Random Geometric Graph
RQ	Research Question
RRT	Rapidly-exploring Random Tree
SA	Situation Awareness
SAT	Situation Awareness-based Agent Transparency
SIPP	Safe Interval Path Planning
SESAR	Single European Sky ATM (Air Traffic Management) Research
SHAP	SHapley Additive exPlanations
SSD	Solution Space Diagram
TCL	Technical Capability Level
TO-AA-SIPP	Time-Optimal Any-Angle Safe-Interval Path Planning
UAS	Uncrewed Aircraft Systems
UAV	Uncrewed Aerial Vehicles
UTM	Uncrewed Air Traffic Management
VFR	Visual Flight Rule
XAIP	Explainable AI Planning
XAI	Explainable AI
XML	Explainable Machine Learning

ACKNOWLEDGEMENTS

Time flies. Before I knew it, four years had passed. This place has become a part of me, filled with countless moments and memories I will always cherish.

I would like to express my special thanks to my supervisors, Clark Borst and Max Mulder. I truly enjoyed every meeting and discussion we had. I met with Clark almost every week, and sometimes our meetings went on far longer than planned because there was always so much to talk about. I deeply appreciate his patience, kindness, and willingness to listen and guide me.

When I had just started my PhD, I accidentally sent a defence invitation to the entire section by mistakenly clicking a link in a colleague's defence invitation email. Even now, I still do not know how it happened, but ever since then, I have been extremely cautious with every link in emails. At that time, Clark replied, "That seems a bit quick." Well, now the defence is finally real!

I remember my first meeting with Max. I was immediately impressed by his enthusiasm. He explained my PhD project in detail and showed me around the lab. I had felt a bit nervous before we met, but I quickly relaxed once the meeting began. Every time I met Max, he greeted me with his characteristic energy – "Hi Yiyuan, how's your life?" – a simple question that never failed to lift my mood.

Beginning my PhD during the COVID-19 pandemic was challenging. It was my first time in Europe, and everything was new to me. However, due to the lockdown, I had little opportunity to explore my surroundings. I am especially grateful to my friends Chengpeng Jiang and Yan Wang, who were there to help me when I first arrived. I also want to thank Sijia Kong and Geqie Sun, who have supported me a lot over the past four years.

I have met so many nice people in our department. I am thankful to my office mates: Wenying Lyu, Yoshinari Hashimoto, Jiayu Chen, Rowenna Wijlens, Gijs de Rooij. It is really great having you around. A special thanks to Liming Zheng for letting me get some hands-on experience with the wind tunnel and his drones. As someone researching drone traffic management, it is really cool to see how drones work up close. I also wish to thank Hang Yu, Yilun Wu, Ziqing Ma, Chaoxiang Ye, Shenqi Wang, Dequan Ou, Moji Shi, Tinghua Li, Xuerui Wang, Sihao Sun, Gang Chen, Yingfu Xu, Bo Sun, Cheng Liu, Yifei Li, Jing Chang, Shushuai Li, Mel, Xiaohuan Lyu, Jingyi Liu, and Yuqian Tu. We had so many fun gatherings, which made my PhD life much more enjoyable.

I would like to thank my colleagues Jan Groot, Malte von der Burg, Aidana Tassanbi, Esther Roosenbrand, Muhammad Fazlur Rahman, Giulia Leto, Haonan Li, Marta Ribeiro, and Junzi Sun. We had a wonderful time at the conferences and symposiums, filled with inspiring discussions and memorable moments. I am also grateful to my colleagues and students Dominik Janisch, Bartłomiej Grochowski, Suyi Wen – working with you is always such a pleasure. My thanks also go to Rene van Paassen, Olaf Stroosma, Annemarie Landman. There was a lot of fun at our "Big" dinner. I would also like to thank Stavrow Bahnam, Robin Ferede, Isabelle El-Hajj, Ece Üreten, and Matthew Yedutenko for their valuable support in my experiments and research.

I also have many friends outside my department, and I want to thank all of you: Xinyu He, Jiarui Zhang, Zhenjie Wang, Yuwei Huang, Qiuju Xue, Jiaxing Fang, Sidi Liu, Jiaxiang Yi, Wencan Wu, Yu Xu, Guanqun Xu, Qiulin Zhu, Zhenxu Qian, Longmiao Gao, Mengjie Zhao, Yujie Wang, Guangze Qin, Xianzhong Liu, Sihan Wang, Yuefan Pan, Jingwei Dong, Xiangyu Yang, Te Tu, Pengtao Gao, Yanghui Xu, and Qin Ou.

This acknowledgement seems to grow into an endless list. I simply want to express my heartfelt gratitude to everyone who has been part of my life. There are still many names I have not mentioned, but I will always remember the times we shared. Finally, I would like to express my deepest thanks to my family and friends in China for their unwavering love, patience, and encouragement. No matter the distance, their support has always been my greatest source of strength.

As this chapter comes to a close, I look back with gratitude and forward with hope. Though it is hard to say goodbye, I am ready to embrace the next phase of my life, carrying with me all the warmth, lessons, and memories of these unforgettable years. I hope we can meet again someday, somewhere. Cheers!

Yiyuan Zou
Delft, October 2025

CURRICULUM VITÆ

Yiyuan ZOU

26-08-1995	Born in Jiangsu, China.

EDUCATION

2014–2018	BE in Air Transportation Management (Air Traffic Control) Nanjing University of Aeronautics and Astronautics, China
2018–2021	ME in Air Transportation Planning and Management Nanjing University of Aeronautics and Astronautics, China
2021–2025	PhD in Control and Simulation, Aerospace Engineering Delft University of Technology, The Netherlands

AWARDS

2018	First Prize of the 15th “HUAWEI CUP” China Postgraduate Mathematical Contest in Modeling
2019	First Prize of the 16th “HUAWEI CUP” China Postgraduate Mathematical Contest in Modeling
2019	China National Scholarship for Graduate Students
2020	First Prize of the 17th “HUAWEI CUP” China Postgraduate Mathematical Contest in Modeling
2023	Third Place in the First TU Delft R Plot-a-thon

List of Publications

7. **Yiyuan Zou**, Clark Borst, *Towards a unified taxonomy for algorithmic transparency: Insights from uncrewed air traffic management*, Cognition, Technology & Work, 1-27, 2025.

6. **Yiyuan Zou**, Clark Borst, *Algorithmic transparency in path planning: A visual approach to enhancing human understanding*, International Journal of Human-Computer Studies 203, 103575, 2025.

5. **Yiyuan Zou**, Clark Borst, *Zeta*-SIPP: Improved time-optimal any-angle safe-interval path planning*, In Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI), pp. 6823-6830, 2024.

4. Dominik Janisch, Suyi Wen, **Yiyuan Zou**, Clark Borst, *Exploring the limits of uncrewed and crewed air traffic segregation by tower controllers*, In 14th SESAR Innovation Days (SIDs), pp. 1-9, 2024.

3. **Yiyuan Zou**, Clark Borst, *Investigating transparency needs for supervising unmanned air traffic management systems*, In 13th SESAR Innovation Days (SIDs), pp. 1-9, 2023.

Under Review

2. **Yiyuan Zou**, Wenying Lyu, Clark Borst, *Solution space path planning for supporting en-route air traffic control*, Submitted to Advanced Engineering Informatics, 2025.

1. **Yiyuan Zou**, Clark Borst, *Exploring the usage of transparency in supervising uncrewed air traffic management systems*, Submitted to International Journal of Human-Computer Studies, 2025.

Uncrewed Air Traffic Management (UTM) is an emerging concept for safely and efficiently managing the growing number of drone operations, particularly near airports. While UTM is expected to rely on high levels of automation to keep drones safely separated from crewed aircraft, human oversight and intervention remain essential to handle unexpected events and contingencies.

To support human supervision of automation, some form of "seeing-into" transparency is required, which reveals the inner workings of automation rather than treating it as a "black box". This dissertation investigates transparent path planning, introducing a visual design method and evaluating its impact within the context of UTM.