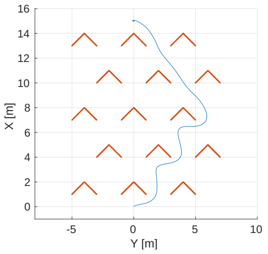
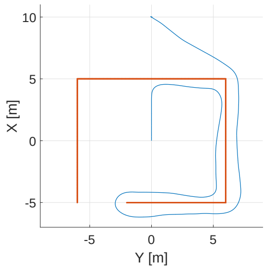
Faculty of Aerospace Engineering



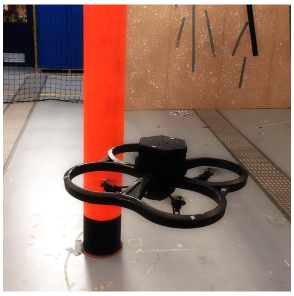


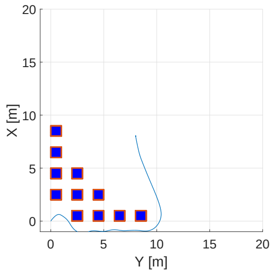
Performance in Obstacle Avoidance

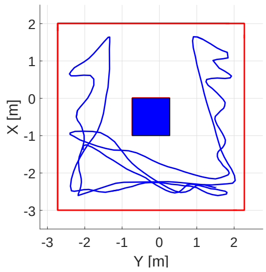
An evaluation study of methods in obstacle avoidance

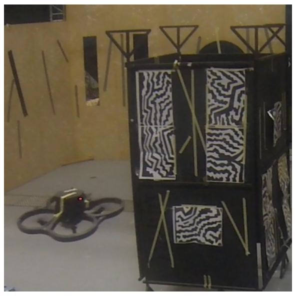
C.W.M. Nous

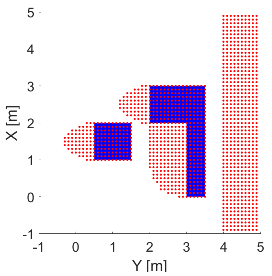
March 20, 2016

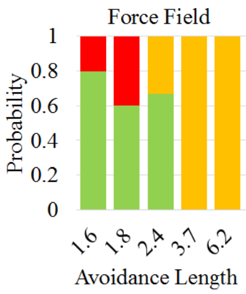


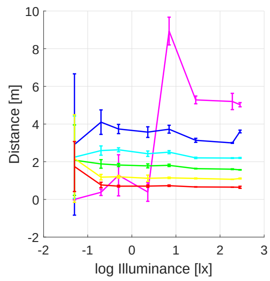












TU Delft Delft University of Technology

Challenge the future

Performance in Obstacle Avoidance

An evaluation study of methods in obstacle avoidance

Master of Science Thesis

For obtaining the degree of Master of Science in Aerospace Engineering at Delft University of Technology

C.W.M. Nous

March 20, 2016

Faculty of Aerospace Engineering · Delft University of Technology

Delft University of Technology

Delft University Of Technology
Department Of
Control and Simulation

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance a thesis entitled **"Performance in Obstacle Avoidance"** by **C.W.M. Nous** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: March 20, 2016

Readers:

Dr. G.C.H.E. de Croon

Dr.ir. C.J.M. Verhoeven

Prof.dr.ir. J.M. Hoekstra

Preface

Dear reader, it is a pleasure to present to you the result of a year of hard work. A year with ups and downs, which brought me across the world. This thesis was written to graduate from the faculty of aerospace engineering at the Delft University of Technology. The work presents a novel method to evaluate obstacle avoidance methods. With the current development rate of new UAVs and the increased automation, the necessity of such method will keep increasing. The topic is very broad and it was sometimes difficult to see the forest from the trees, but I believe the work is a step forward to see the forest more clearly within the field of obstacle avoidance.

The work consist of five parts. The first part is the main thesis paper. The second part is a conference paper, which can be seen as a short version of the full work in which only the key contribution are explained. For a reader under time pressure this would be the best part to read. In the third, fourth and fifth part additional information is provided. Especially the overview tables are interesting for those readers looking for a broad scope of the field of obstacle avoidance.

I would like to thank the people how helped me in the development of this thesis. In this first place Guido de Croon, who was my main supervisor, the weekly meetings and countless discussions helped improving the quality of the work. Also the trust and positive attitude served as a great motivation. I would also like to thank the staff of the MAVLAB, particularly Roland Meertens, Erik van der Horst, Cristophe de Wagter, Sjoerd Tijmons and Kimberly McGuire. Who helped with many practical issues to get the UAV flying, but also with many brain-storm sessions. Finally I would like to thank my fellow students who were always prepared to share some misery, when thinks did not go as planned.

In the end I learned a lot during this research, from the practical skills of working with drones to the academic approach of doing research. Hopefully the reader will find some interesting thinks and not feel to much reluctance in reading the work, I wish you all the best.

Clint Nous
March 20, 2016

Contents

Acronyms

APD	Average Pass-through Distance
AR	Avoidance Ratio
ATM	Air Traffic Management
CAS	Collision Avoidance System
CC	Collision Cone
CDR	Collision Detection & Resolution
DCE	Distance to Closest Escape
DP	Dynamic Programming
FN	False Negative
FOV	Field Of View
FP	False Positive
FPS	Frames Per Second
IP	Image Processing
IR	Infrared Sensor
MAV	Micro Air Vehicle
MPC	Model Predictive Control
OA	Obstacle Avoidance
PIR	Passive Infrared Sensor
RCS	Radar Cross Section
RMS	Root-Mean Squared
ROC	Receiver Operating Characteristic
RRT	Rapidly exploring Random Tree
SAD	Sum of Absolute Differences
SLAM	Simultaneous Localization and Mapping

SNR	Signal to Noise Ratio
TCAS	Traffic Alert and Collision Avoidance System
TN	True Negative
TP	True Positive
UAV	Unmanned Aircraft Vehicle System
VFF	Virtual Force Field
VFH	Vector Field Histogram

List of Figures

List of Tables

Part I

Thesis paper

Performance Evaluation in Obstacle Avoidance

Clint Nous
Delft University of Technology

***Abstract*—No quantitative procedure currently exists to evaluate the obstacle avoidance capabilities of robotic applications. Such an evaluation method is needed for comparing different methods, but also to determine the operational limits of autonomous systems. This work proposes an evaluation framework which can find such limits. The framework comprises two sets of tests: detection tests and avoidance tests. For each set, environment and performance metrics need to be defined. For detection tests such metrics are well known. For avoidance tests however such metrics are not readily available. Therefore a new set of metrics is proposed. The framework is applied to a UAV that uses stereo vision to detect obstacles and three different algorithms to calculate the avoidance manoeuvre.**

***Index Terms*—Obstacle Avoidance, Evaluation Framework, Complexity Metrics, Benchmark**

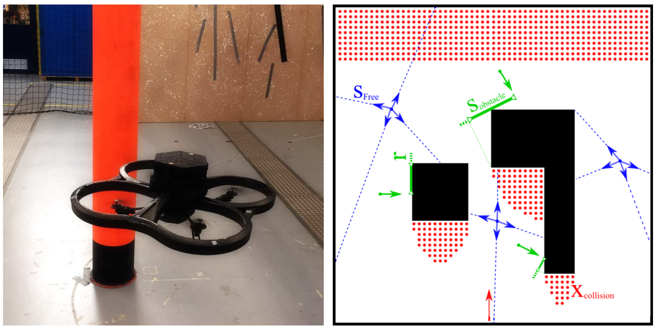


Fig. 1: The framework is applied for an ARDrone with an omnidirectional stereo vision system (left). In the framework environment complexity metrics are introduced, which provide a guideline for tests in different environment conditions (right).

I. Introduction

Autonomous flight with aerial robots has many promising applications, such as surveillance, inspection or package delivery. To carry out such tasks a reliable obstacle avoidance system is essential. Many obstacle avoidance systems are available [1], but it is unknown what the performance of these systems is. No quantitative evaluation procedure to measure the performance of obstacle avoidance systems is available.

Such a procedure is required to determine the reliability of obstacle avoidance systems and to find in what conditions these systems can safely operate. Knowledge of these operational conditions is crucial when deploying UAVs for practical applications.

The functioning of an obstacle avoidance system is often demonstrated in a single environment. Environments found in literature are diverse, and include: forests [2], buildings [3], hallways [4] and sparse obstacle courses [5]. It is difficult to make performance predictions based on a single environment since the performance of an obstacle avoidance system depends on the environment in which it operates. For the same reason it is difficult to compare obstacle avoidance methods which are demonstrated in different environments.

Therefore a standard evaluation procedure is needed. Such procedure allows us to identify strengths and weaknesses of an obstacle avoidance system. It could provide a quantitative measure of the performance of an obstacle avoidance system, which could be used to compare obstacle avoidance methods or to determine the state of the art in the field of obstacle avoidance. Without a good evaluation method the development of new algorithms are likely to lead to ad-hoc solutions, this is currently seen in the field of obstacle avoidance.

An attempt to create such an evaluation method is proposed by Mettler et al. [6]. In this method six simple obstacle courses and an urban environment are used to test the performance of an obstacle avoidance algorithm. Unfortunately Mettler does not motivate the choice for the selected tests and no explanation is given on how representative these tests are for the overall performance of a system.

A different approach is followed by Kuchar et al. [7]. Kuchar describes an evaluation method based on the verification method of TCAS systems. Millions of simulations are run to encapsulate all possible scenarios which could occur during the life-time of an aircraft. For a small UAV however, the environment can be much more complex. It is therefore infeasible to run simulations for each scenario.

Besides this work and papers in which detection and avoidance methods are compared [8], [9], researchers have not attempted to quantify or benchmark the performance of obstacle avoidance algorithms. This is remarkable, considering the countless research contributions done in the field. Such evaluation methods and benchmark data sets are common practice in other research fields such as computer vision or control engineering [10].

In this paper an evaluation framework is proposed, which makes it possible to quantify the performance of an obstacle avoidance system. A key aspect of this framework is to quantify the environment using specific metrics. The evaluation framework and its metrics are discussed in section II to V, which is applied to a robotic application in section VI to XI. The framework is evaluated and concluded in sections XII & XIII.

II. Evaluation Framework

Developing a standardized evaluation method is difficult due to three aspects:

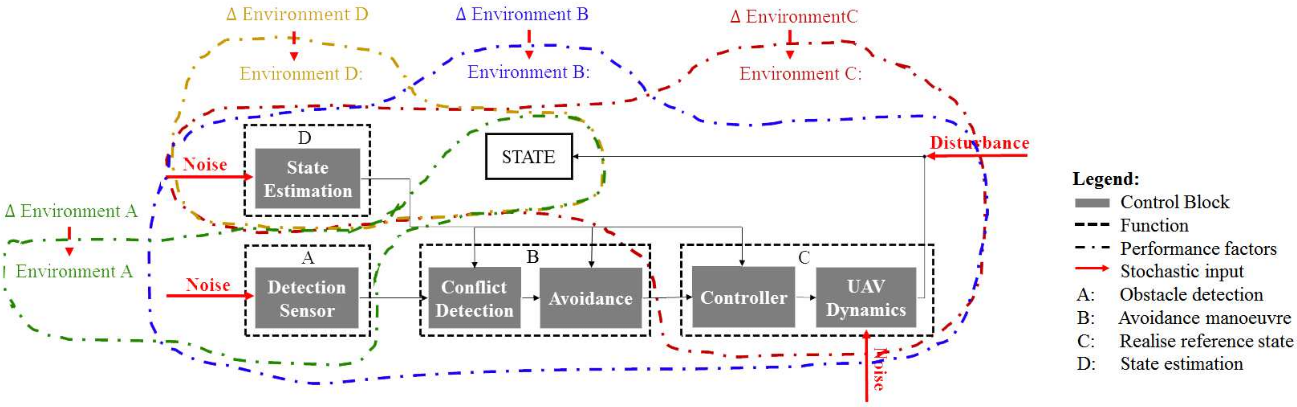


Fig. 2: Obstacle avoidance control loop in which the performance dependencies of four functions(A-D) are visualised.

1) Performance depends on the complete control loop
2) There is a high variety of operational conditions
3) Each obstacle avoidance method is developed for a different set of environments.

The first difficulty is caused by the relation between the obstacle avoidance method and the UAV on which it is applied. No direct comparison can be made between methods applied on different UAVs, since the performance is dependent on the platform. The second and third challenges are caused by the performance dependency on the environment. These environment conditions are diverse and sometimes difficult to quantify. Since each method can be sensitive to different environment conditions no 'one size fits all' test set-up is possible. Different tests are needed for different methods.

The first challenge is faced by analysing the entire obstacle avoidance control loop and by identifying which factors influence the functions in this loop. The complete control loop of an obstacle avoidance system is shown in Figure 2. In the figure four main functions are identified (A-D), which are shown by dotted squares. For each function a 'cloud' is drawn to visualise the factors that impact the performance of each function. It specifies in which way the performance depends on the control loop.

The first function (A) is detection, which consist of the detection sensor. Function B is the calculation of the avoidance manoeuvre, which is a combination of conflict detection and avoidance. One could argue to evaluate conflict detection and avoidance separately, but since the two are often strongly coupled (conflict detection serving as a trigger for avoidance) and have the same goal (creating a control reference), the choice is made to evaluate the blocks as one entity. The third function is the realisation of the reference state, which is a combination of the controller and the UAV dynamics, shown by function C. The final function is the state estimation represented as function D.

This paper focuses on the obstacle detection function (A) and the calculation of the avoidance manoeuvre (B), these are the primary functions of an obstacle avoidance system. The factors which influence the performance of the detection function are visualised by the green cloud. It can be seen that three factors are present: the state, sensor noise and the environment. The state specifies the distance but also the velocity with respect to an obstacle, both influence detection performance. The noise arrow represents the internal noise of the sensor and has a direct impact on the measurement. Finally the environment conditions to which the detection sensor is sensitive is represented by 'Environment A'.

The factors that influence the performance of function B is shown by the blue 'cloud'. All blocks in the control loop effect this performance. A bad detection for example increases the chance of colliding with an obstacle, the same holds for a bad state estimation. Also three types of noise and a disturbance are included. Lastly the environment for which the performance of the avoidance manoeuvre is sensitive is included and represented as 'Environment B'. For completeness also the 'clouds' of function C and D are drawn.

Now an overview is presented of what affects the performance of obstacle avoidance functions, it can be used to define evaluation tests. Each factor in the clouds of function A and B can be used as an independent variable in a performance test. In this paper the focus is put on the environment factors. The others factors are assumed to be constant. These constants should be clearly explained since the performance is dependent on them.

To quantify the environment factor, (such that it can be used as an independent variable) it needs to be described as a metric. When the environment is not described using specific metrics it becomes difficult to make performance predictions. The definition of these metrics will be discussed in the next section. Besides the independent variable a dependent variable is needed to quantify the performance. So far it has not been discussed how performance is defined. This can be done by specifying a performance metric. Both metrics for the detection (A) and avoidance (B) function are discussed in the next sections.

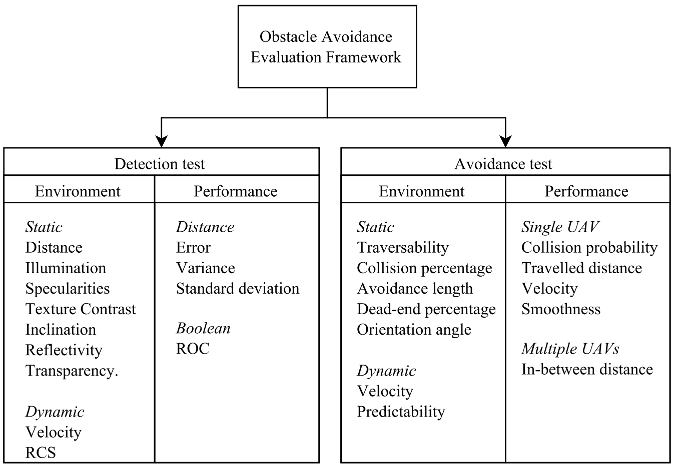


Fig. 3: Overview of possible environment and performance metrics structured by general characteristics.

III. Performance and Environment Metrics

Environment metrics define the environment in which an experiment is done while the performance metric specifies what is measured in this environment. The environment and performance metrics selected for the evaluation measurements depend on the obstacle avoidance system that is used. Therefore a broad overview of the field is required. Several surveys have been conducted [1], [11]–[14]. For each method in these overviews a set of relevant metrics should be selected to determine the detection and avoidance performance.

For the detection function these metrics depend on the sensor and its processing, for the avoidance function these metrics depend on the avoidance algorithm. Also some general characteristics influence the type of metric. For instance, an obstacle avoidance task with static obstacles requires different environment metrics than an obstacle avoidance task with dynamic obstacles. In the dynamic case, additional factors such as the velocities and predictability of the obstacles play an important role. he characteristic whether the task has the UAV move in 2D or 3D has a similar influence on the metrics, as 2D environment metrics may not directly generalize to the 3D case.

Also the performance metrics are influenced by the general characteristic. For the detection function a different performance metric is required when only a boolean variable is measured than when a distance measurement is performed. Also the amount of UAVs can have an influence on the performance metric, extra metrics such as the in-between distance could be used for example. This general division is shown in Figure 3. The distinction between 2D and 3D is omitted in the figure.

Although the framework is applicable to all these cases, the focus in this paper will be on metrics for a 2D environment with static obstacles, a single UAV and a detection sensor that measures the distance to an obstacle. In the following the specific metrics for the detection and avoidance tests is discussed.

IV. Detection tests

The goal of the detection tests is to determine under what conditions obstacles can be detected. To define the metrics for these tests a general idea of the available methods is required. Six main detection sensors can be identified: monocular-vision, stereo-vision, infrared, ultrasonic, laser and radar. In this discussion cooperative sensors such as ADS-B are omitted. First the relevant environment metrics for these sensors are discussed and thereafter the performance metrics.

A. *Environment metrics*

Each sensor is sensitive to different environment characteristics. Fortunately these are fairly well known. An overview of these metrics can be seen in Table I. In the table the relevant metrics for each sensor are specified. The distance, for example, is relevant for all sensors but the illumination is only relevant for monocular and stereo vision. This is not and extensive list and could be complemented by researchers using this framework. It should be noted that, even within a type of sensor, differences are present. For example in stereo vision different metrics could be needed for different types of matching algorithms. This table can be used to select the relevant environment metrics for the performance tests.

TABLE I: Relevant environment metrics for detection sensors.

	Monocular vision	Stereo Vision	Infrared	Sonar	Laser	Radar
Static						
Distance	✓	✓	✓	✓	✓	✓
Reflectivity	✓	✓	✓	✓	✓	
Illumination	✓	✓				
Texture Contrast	✓	✓				
Texture Angle		✓				
Illumination (Infrared)			✓			
Inclination			✓	✓	✓	
Transparency	✓	✓			✓	
Dynamic						
Velocity	✓	✓	✓	✓	✓	✓
RCS						✓

B. *Performance metrics*

Performance metrics often seen in literature for detection sensors are the distance error and variance. Another metric is the receiver-operating-characteristic curve, in which true positives and false positives are plotted as function of a threshold. This plot is particular useful for a binary detection sensor. A third performance measure seen in literature is the computational effort, for which the metric of frames per second or computation time is used. Since the computational effort does not depend on the environment, the metric is not seen as a performance metric.

V. Avoidance tests

The goal of the avoidance tests is to determine under which conditions detected obstacles can be avoided. Again metrics need to be selected which are dependent on the method that is being used. Unfortunately no simple division can be made between the wide diversity of avoidance methods. Methods vary from simple rule based instructions to complex path planners. Even within path planning a large variety exist. All these methods can be sensitive to different environment conditions. A similar table as Table I needs to be constructed in which the columns represent the methods and the rows the environment metrics. But from literature it cannot be determined what these environment metrics should be. Therefore a novel set of such metrics is proposed.

A. Environment metric

For detection sensors the relation between environment and performance is often discussed in literature, but for avoidance it is not. As mentioned in the introduction, most research contributions use a specific environment to test their algorithm, without information on how representative these tests are. Therefore not much is known about the performance of these algorithms in different environments.

Only a few environment metrics are seen in literature: the width of the obstacles, in-between distance or the density [15], [16]. These metrics do not take the size of the UAV into account, while this is essential. It is more challenging for a UAV with a radius of 0.5 m to fly trough obstacles with an in-between distance of 1.0 m, than it is for a UAV with a radius of 0.1 m. In the following a new set of non-dimensional environment metrics is proposed which take these properties into account. The following five metrics are proposed: 1) Traversability, 2) Collision state percentage, 3) Average avoidance length, 4) Dead-End percentage and 5) Average orientation angle. This is not an exhaustive list and could be complemented by other researchers.

1) Traversibility: The fist metric is the traversability which is related to the obstacle density. An obstacle avoidance task becomes more difficult when the distance between obstacles becomes smaller. This difficulty is quantified by the traversability.

The traversibility could be represented by dividing the amount of occupied space by the total flight space (i.e. density). But such metric would not be able to make a distinction between an environment with multiple small obstacles and one large obstacle. The density metric used by [16] is also possible, in which the amount of obstacle per square meter is used. But also this metric can be questioned since a room with five small obstacles is less challenging than a room with five large obstacles.

Another option, which is independent of the flight space, is to look at the distances between neighbouring objects. The average of these distances gives an indication about the difficulty of the avoidance task.

In this paper a traversibility metric is selected which is calculated by selecting random positions and headings in a flight space. For these positions and headings the maximum straight-flight distance s is determined. In Figure 1 the blue lines display these distances for four points and four headings. The average of these distances over n samples gives a measure of how densely packed the environment is and therefore how challenging the performance task is. The calculation is shown by Equation 1. In the equation the average value is divided by the radius of the UAV (r), to make the metric non-dimensional.

$$TRAV = \frac{1}{n \cdot r} \sum_{0}^{n} s_{passageway} \qquad (1)$$

2) Collision state percentage: The second metric is the collision state percentage. This metric combines the dynamical constraints of the UAV with the available flight space. Flying inside a room with obstacles becomes more difficult when the turning radius of the UAV increases, but also when the size of the room decreases. This effect is quantified by calculating the percentage of states for which a collision is unavoidable. To determine these states each state is propagated into the future. When all propagated trajectories lead to a collision, the state is marked as a 'collision state'.

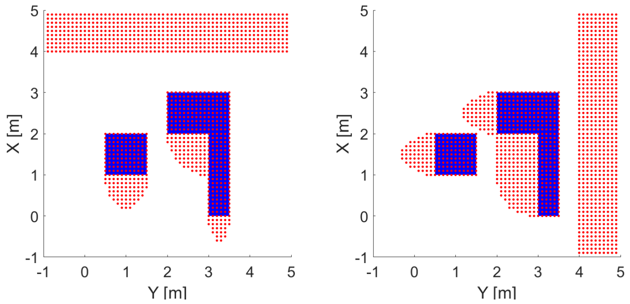


Fig. 4: Visualization collision state factor; $\psi = 0$ (left), $\psi = 0.5\pi$ (right).

The propagated trajectories are calculated using a minimum turning radius which depends on the UAV dynamics and the velocity. Because the turning radius is influenced by this reference velocity it is considered as a constant. Also the reaction time (caused by the detection frame rate, calculation time or system delays) is taken into account when calculating the trajectories.

The states selected for the 2D case are the position (x, y) and the heading (ψ). The velocity is assumed to be constant and therefore omitted. The yaw, roll and pith angles/rates are omitted from the state space as well, it is assumed that these have a negligible effect on the collision ratio. An example can be seen in Figure 4. The collision positions are shown by the red dotes, for headings of $\psi = 0$ and $\psi = 0.5\pi$.

3) Average avoidance length: The third proposed metric is similar to the size of the obstacle. This metric quantifies the difference between a forest environment with thin obstacles and a building environment with wide obstacles.

The avoidance length could be specified by dividing the length of the avoidance path, by the length of the direct path. This ratio gives an idea of how challenging the avoidance task is. Since this ratio is very sensitive to the initial and goal state a different approach is used.

In this paper the average avoidance length is calculated by averaging the needed lateral movement to avoid obstacles at each time-step during a flight. The lateral movement is the sum of the (projected) width of the obstacle and the radius of the UAV, which is shown by the green lines in Figure 1. Again the metric is made non-dimensional by dividing it through the radius of the UAV (Equation 2).

$$AAL = \frac{1}{n \cdot r} \int_0^n s_{escape} dt \tag{2}$$

4) Dead-end percentage: The fourth metric is the dead-end percentage. Dead-ends increase the chance of getting stuck in a local minimum and therefore influence the overall performance of the algorithm. A dead-end is defined as a point for which a heading change higher than 0.5π is required to reach the goal. This is visualized for two 2D cases in Figure 5.

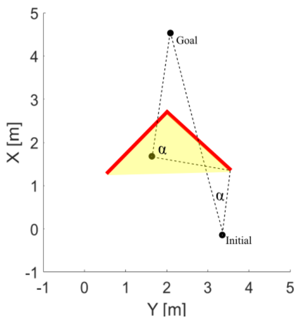


Fig. 5: Visualization of the dead-end area (yellow) for two different obstacles (red).

The ratio of the dead-end area ($\alpha > 0.5\pi$), with respect to the total flight area is used to quantify the level of difficulty.

5) Average orientation angle: The final proposed metric quantifies the orientation of the obstacle(s) relative to the UAV. One can imagine that it is simpler to avoid an obstacle at a certain angle, than an obstacle perpendicular to the flight-path. This metric is calculated by taking the average of the angle between the tangent plane of the obstacle and the flight path angle of the UAV.

B. Performance metrics

Besides the environment metrics also performance metrics are needed. Generally three types of performance metrics are seen in evaluating avoidance algorithms: computational time, success rate and path optimality. There exists a hierarchy between success rate and path optimality, since path optimality can only be determined when an obstacle is successfully avoided. Path optimality metrics can therefore be seen as secondary metrics.

For the success rate three scenarios are distinguished: successful flights which reach their goal safely, unsuccessful flights which do not reach the goal but do not collide either and flights which lead to a collision. The primary performance metric specifies the percentage for each scenario.

For the secondary performance metrics several suggestions are made by Mettler et al. [6]: duration of flight, velocity, energy usage, path smoothness and obstacle clearance. In this paper the choice is made to focus on two optimality metrics: travelled distance and average velocity. The duration and energy usage can be derived from these metrics.

VI. Robotic Serving Application

In the following sections the previously discussed framework is applied to a robotic application which is shown in Figure 1. The shown UAV makes use of six stereo cameras to detect obstacles and has a choice between three different avoidance strategies. The chosen general characteristics for the application are: 2D, static obstacles, single UAV with distance measurements. The detection and avoidance performance are discussed in the following sections.

VII. Detection Performance

The six stereo cameras have a horizontal field of view of $57.4°$, which create an almost complete $360°$ view of the environment. The cameras have a baseline of 60 mm, a focal length of 118 px and weigh only four grams [17] . Two small images of 96×128 px are produced. The pixels are sparsely matched on the stereo board. The disparity map is created using the sum-of-absolute-difference scheme presented in [18].

To evaluate this detection sensor, Table I can be consulted. In the table five static tests are suggested for stereo vision. Four of these tests are presented in the following subsections.

A. Distance

For the first test the distance is selected as independent variable. The performance effect of the distance is measured by placing an obstacle of 1×1 m (Figure 6) in front of the stereo-camera. The distance to this obstacle is varied from 0.5 m to 4 m. The results can be seen in the right of Figure 6.

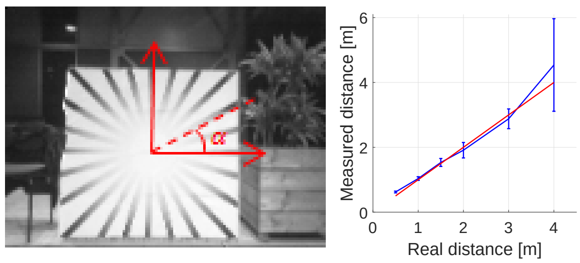


Fig. 6: Test obstacle (left); Results distance test, red = groundtruth, blue = stereo measurements (right)

The figure shows that an obstacle can be measured accurately up to three meters. For larger distances the standard deviation increases rapidly. At four meters a standard deviation of more than 1 m is present. This increase in variance is fundamental to stereo vision [19]. Though a smaller variance is possible. Theoretical a resolution of 0.38 m is possible at 4 m distance.

B. Illumination

The second selected environment metric is the illumination. Measurements were conducted in a theatre in which the light could be controlled. Tests in an illumination regime from 284 lx to 0.05 lx were conducted.

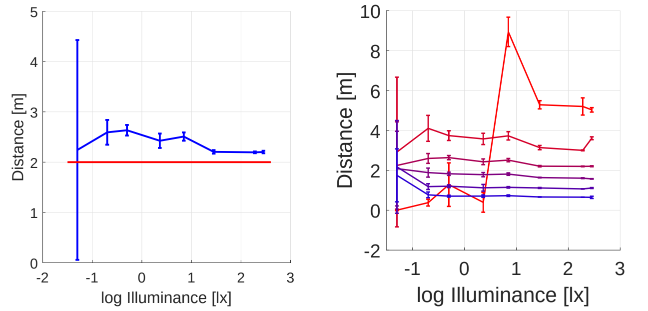


Fig. 7: Results illuminance tests at a distance of 2 m (left) and for several distances (right), 0.5 m = blue to 4 m = red.

The distance measurements of an obstacle at 2 m are shown in the left of Figure 7. A decrease in illuminance results in a small increase in detection error, but it remains remarkable accurate up to an illumination of $10^{-0.7}$ lx. For illuminances lower than $10^{-1.3010}$ lx no accurate distance estimates can be acquired.

The measurements are repeated at different distances, for which the results are shown in the right of Figure 7. A similar behaviour can be observed for distances from 0.5 to 3 meters.

A different behavior is seen at 4 m. At an illumination of $10^{0.85}$ lx the distance estimate increases up to a value of 9 m. When the illuminance is lowered further the distant measurement drops to a value of 0.4 m. So suddenly an obstacle far away is detected to be very close. This has a drastic impact on the behaviour of the avoidance manoeuvre.

Fig. 8: Left stereo image (Left), right stereo image(Right)

The reason of this large underestimation can be seen in Figure 8. The images from the left and right cameras of the stereo vision system are shown. It can be seen that the illuminance in the left image is lower than in the right image. Due to this difference, wrong pixel matches are found. This effect increases when the illuminance is decreased.

To assure that a similar amount of illuminance is present in both images the light should be evenly distributed. The relation between the distribution of light and the illuminance depends on the light sources in the room.

For the theatre a minimum illuminance of 7 lx was needed to get an even light distribution. But this number could be lowered by increasing the number of light sources. Most rooms will have less light sources than the ones present in the theatre, so generally an illumination higher than 7 lx is required. A better understanding is required of the relation between illuminance and the light distribution to validate this conclusion.

C. *Texture contrast*

The third metric which is analysed is texture contrast. One 'ray' in Figure 6 is analysed. In each ray the grey-scale is decreased from white in the centre to black at the edges. By doing this the contrast increases from zero to a maximum between black and white.

The distance measurement of the matches found inside the red area, shown in Figure 9, are analysed. In the right of the figure it can be seen that from a certain amount of contrast the stereo algorithm is able to find a match. From this switching point the distance measurement stays relatively constant.

So the contrast does not influence the accuracy of the measurement but rather the point at which an obstacle is detected. This 'switching point' occurs halfway the normalized contrast scale.

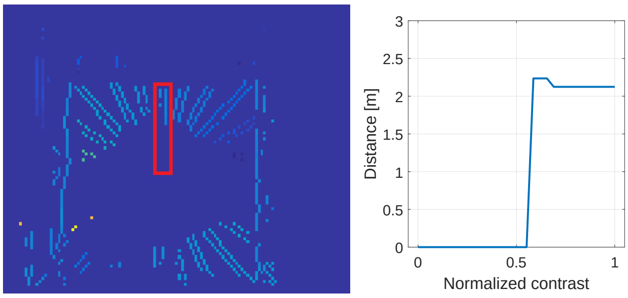


Fig. 9: Disparity map of an obstacle at 2 m (left); Distance measurements for different amounts of contrast (right).

The amount of contrast in an image is highly dependent on the amount of light. It would therefore be interesting to see how this 'switching point' changes when the amount of illumination is changed.

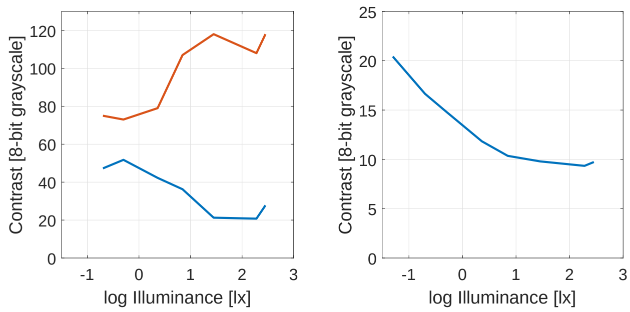


Fig. 10: Contrast vs illuminance test, blue = required contrast, red = maximum contrast (left); Average contrast in image (right).

This relation is shown in Figure 10. The red line in the figure shows the difference in 8-bit gray-scale value at the edge of the panel(maximum contrast). This contrast decreases when the illumination is decreased. The blue line shows the 8-bit gray-scale value at which a first pixel match is found. A minimum contrast of 20 8-bit gray-scale is required to find a match. This value increases when the illuminance is decreased. This effect can be explained by the increase of noise in the image. This noise can be quantified by the average contrast, which is shown in the right of Figure 10. A match is only found when a pixel 'stand outs' from this noise. Therefore a higher contrast is needed to find a disparity.

D. Texture angle

The final selected metric is the texture angle. This performance is measured by comparing the measurements form the different 'rays' in Figure 6. The angle is measured with respect to the horizon as shown by α in the figure.

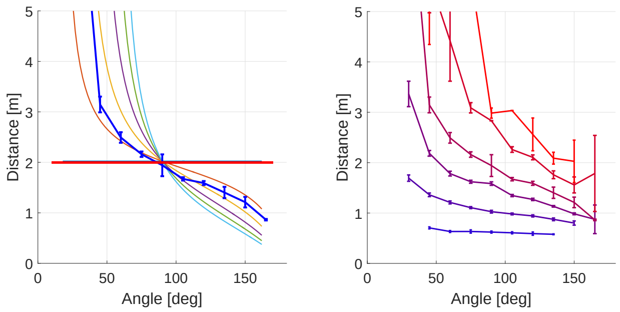


Fig. 11: Results texture angle test for an obstacle at 2 m, with values of a theoretical offset (left); Results texture angle test for different distances (right).

The distance measurements for an obstacle placed at 2 m is shown in the left of Figure 11. A drastic effect of the texture angle can be observed. The stereo-camera is not able to detect texture angles up to 30°. For angles between 30° and 75° an overestimation of the measured distance is present. For angles higher than 110° an underestimation of the object is present.

This effect can be explained by the vertical alignment of the left and right image. When the two images are misaligned a match for an inclined obstacle is found a few pixels later or earlier (depending on the direction of the misalignment). The theoretical effect of this misalignment is also plotted in the left figure. Five theoretical plots are shown with a horizontal misalignment ranging form 1 to 5 px. It can be seen that the measurements lie in-between the theoretical line of 1 and 2 px. The stereo-camera should be corrected for this horizontal misalignment.

Again the measurements are performed at different distances. The results can be seen in the right of Figure 11. The figure shows a similar effect for all distances, the effect enhances when the distance is increased.

VIII. Avoidance Methods

Now that an idea of the detection performance is present, the performance of the avoidance manoeuvre can be analysed. Three avoidance strategies are applied, which are discussed in the following subsections.

A. Force field method

The force field method calculates the avoidance manoeuvre by replacing an obstacle by a force. The sum of the forces is used by a control law to determine the control reference. These forces and control laws can be be defined in many ways. In this paper a method similar to Kandil et al. [20] is implemented. The definition of these forces and the control law is given in Equation 3.

$$F_{total} = k_o \cdot erf(d_g) + k_g \sum_i \frac{C_i}{d_i}^2 \tag{3a}$$

$$F_{rep,i} = \left(\frac{C_\alpha(\alpha_i) \cdot C_v(v_i)}{d_i}\right)^2 \tag{3b}$$

$$C_i = F1 + F2 \cdot cos(C_{freq} \cdot \alpha_i) \cdot erf(d_g) \tag{3c}$$

Two forces are defined, a goal force and an obstacle force. The goal force is a function of the distance to the goal, the obstacle force is depended on the relative distance d_i and a factor C_i. Several constants are present which can be tuned. An advantage of this adaptability is that many behaviours can be realized. In this paper the following constants are used: $K_o = 1$, $K_g = 1$, $F1 = 0.1$, $F2 = 0.9$, $C_{freq} = 1$, $V_{max} = 1$ m/s.

B. Potential field method

The second avoidance strategy is similar to the first but uses potential fields instead of vectors. The control law is based on a method from Huang et al. [21] and shown in Equation 4.

$$\ddot{\phi} = -b\dot{\phi} - k_g(\phi - \psi_g)(e^{-c_1 d_g} + c_2) + ... \tag{4a}$$
$$... \sum_i k_{o_i}(\phi - \psi_{o_i}))(e^{-c_3|\phi - \psi_{o_i}|})(e^{-c_4 d_{o_i}})$$

$$v = v_{max} \cdot e^{-k_v \psi_o} \cdot erf(0.5 d_g) \tag{4b}$$

In Equation 4a the angular rotation is shown as function of three terms. The first term is a damping term. The second term is the goal term, similar to the goal force of the previous method. The third term is the obstacle term which is the sum of the derivatives of the obstacle potentials. The obstacle potential is dependent on the distance and heading of the obstacle. A second control law calculates the velocity reference. This law is shown by Equation 4b. The velocity reference is a function of the distance and total potential of the obstacles. Also for this method many constants can be tuned, the following constants are selected: $k_{obst} = 40$, $k_{goal} = 2$, $c_1 = 0.4$, $c_2 = 0.4$, $c_3 = 5.0$, $c_4 = 0.5$, $b = 0$, $v_{max} = 1$, $k_v = 1$, $\epsilon = 0$.

C. Rule based method

The third method is a rule based method, which uses simple rules to determine the avoidance manoeuvre. In this method no velocity reference is created but only a 'safe' heading. The logic used to determine this heading is show in Figure 12.

The first check in the diagram is to see if an obstacle is present. If an obstacle is detected for the first time a

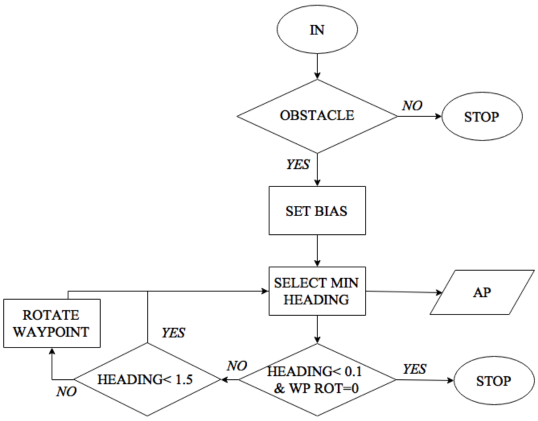


Fig. 12: Rule Based Logic

left or right bias is selected. After this bias is selected the heading closest to the goal heading is chosen. This simple method would be sufficient if no dead-ends are present in the environment. To also be able to avoid dead-ends extra logic is added, this logic is shown in the left bottom of the figure. The method with and without the added logic can be seen in Figure 13.

The code of the three avoidance methods can be found on GitHub[1]. The following section analyses these three methods using the evaluation framework proposed in section II.

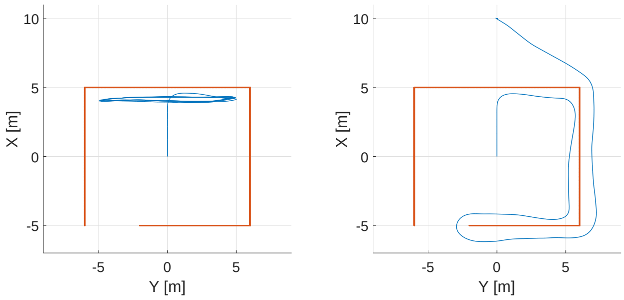


Fig. 13: Rule based avoidance method with and without added logic

IX. Avoidance Assumptions

For these avoidance methods it is not clear which metrics are relevant. No equivalent table as Table I is available. Therefore all proposed metrics in section III are used to determine the avoidance performance. The measurements can be performed using a simulation or by doing real-flight tests. In this paper both measurements are performed. In both cases it is important to state the assumptions made in the complete control loop as described in section II. Each aspect in the 'avoidance cloud' in figure 2 should be clarified before performing a test. These assumptions are discussed first after which the results of the performance tests are presented

A. Assumptions real-flight

The measurements for the real-flight tests[2] were performed on an AR.Drone 2.0 using the paparazzi autopilot. The control loop starts with detection, for which the six stereo boards analysed in section VII are used. The detection noise is estimated to have a standard deviation of 0.05 m. The avoidance methods were explained in the previous section. The state realisation is executed using an INDI attitude controller [22] with a PID velocity controller. This results in real states through the dynamics of the AR.Drone. The states are estimated using an Optitrack system, the state noise is therefore assumed to be negligible. The actuator noise of the AR.Drone is present but is assumed to be negligible as well. There were no wind gusts in the flight arena, but some disturbances were present due to the airflow created by the UAV itself.

B. Assumptions simulation

When performing simulations more options are available for testing the performance. In a simulation the dynamics could be changed or extra noise could be added for example. Also more simplifying assumptions can be made such as perfect detection or zero delay. Simulations also give practical advantages, a wide variety of environments could be tested in a simulation which are not feasible in real-life.

The simulations presented here attempt to replicate the real-flight test. To do this a detection sensor is used which has a standard deviation similar to the sensor used for the real-flight test. The avoidance methods are the same as for the real flight test. The only difference are some tuning parameters. The following value is changed for the vector method: $K_o = 1.75$. For the potential field method the following values are changed: $k_{obst} = 12$, $k_{goal} = 4$, $k_v = 0.1$. This different tuning is needed since a different controller is used in the simulation. An outer loop PID controller is used. The inner loop control is embedded in the model. A perfect state estimation is assumed and no disturbances are present.

X. Avoidance Performance

The only aspects not discussed in the previous discussion is the environment. In the following subsections the performance in different environments is analysed using the metrics discussed in subsection V-A.

A. Traversibility

The first metric is the traversibility. The performance under five different traversability values is tested. This is done by decreasing the distance between 1 m square obstacles. The distance is changed from 3 m down to 1 m. The environment for the highest and lowest values are shown in Figure 14.

For the real-flight tests three obstacles where used for which the in-between distance was changed in the same way. A snapshot of one of the runs and a top-view can be seen in Figure 15.

For each traversibility value, five flights were performed. The result of the real and simulated tests can be seen in Figure

[1] github.com/paparazzi/tree/master/sw/airborne/modules/obstacle_avoidance

[2] youtube.com/playlist?list=PL_KSX9GOn2P8DI_viJWg_hpG-uBLP-yNK

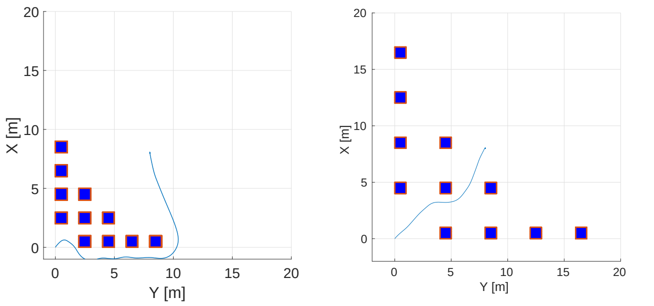


Fig. 14: Simulation environment with a high and low density.

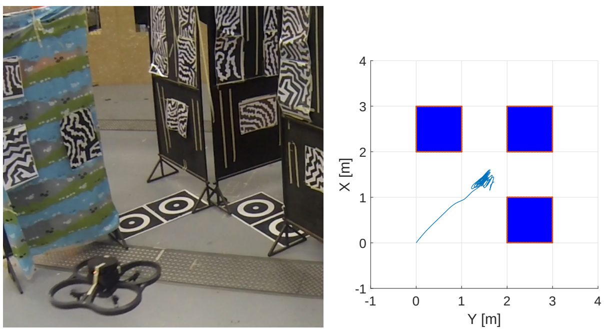
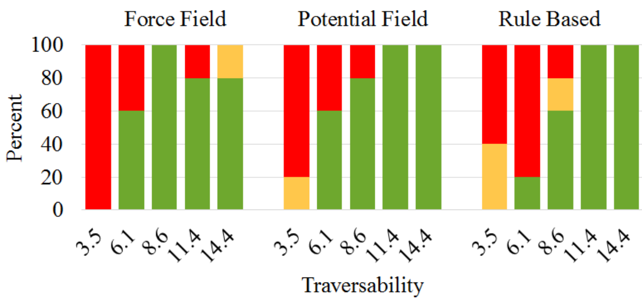
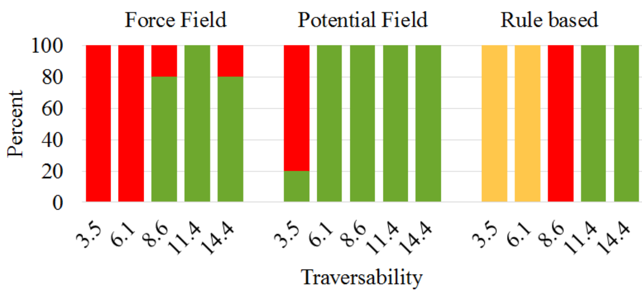


Fig. 15: Snapshot (left) and plotted top-view (right), of a traversability performance test.



(a) Real-flight tests

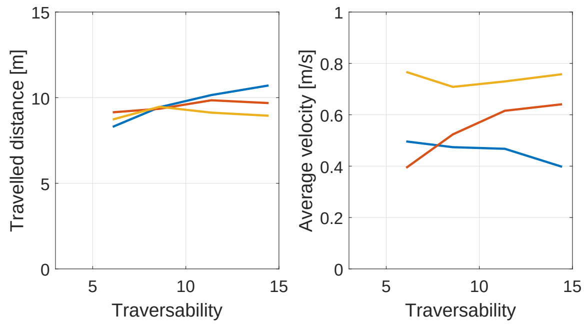


(b) Simulations

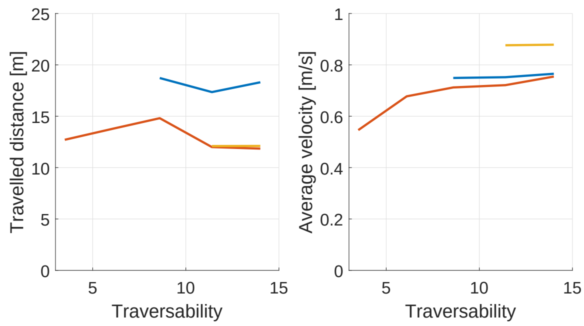
Fig. 16: Percentage of flights that reach the goal (green), do not reach the goal but do not collide (orange) or result in a collision (red).

16. For the real-tests it can be seen that the three methods are not able to reach the goal for a traversability of 3.5. From a traversability of 6.1 some runs reached their goal, but still a collision percentage of 0.4 is present for the force and potential field method. For the rule based method this probability is even higher. For a traversability higher than 8.6 most runs are successful.

The simulation results show a bigger difference between the three methods. The rule based methods is less successful at low traversability values. This is caused by the relative high 'safety buffer' which is used to select a safe heading.



(a) Real-flight tests



(b) Simulations

Fig. 17: Path optimality for force field (blue), potential field (Red) and Rule based (yellow) method.

The secondary performance metrics are shown in Figure 17. The travelled distances are similar for each method for the real-flight tests. For the simulations the force field method shows a higher travelled distance than the other methods. This is caused by several oscillations in the path. The reason less oscillations are present for real-flight tests can be explained by the non-homogeneous texture of real obstacles. This results in unsymmetrical forces, forcing the UAV to fly in a certain direction.

When looking at the average velocity, it can be concluded that the rule based method is the fastest method. The force field method is the slowest. The figure also shows an increase in the average velocity of the potential field method. This can be explained by Equation 4b. In the equation the reference velocity is a function of the sum of the potential fields. For a dense environment the obstacles are closer to the UAV and therefore have a higher potential causing the reference velocity to be lower. This decreases the collision rate but also decreases the optimality of the path. The average velocity of the vector field method and the rule based method are not influenced by the traversibility.

B. Collision state percentage

The second metric is the collision state percentage. The performance under different collision state percentages is measured by decreasing the room in which the UAV flies. To prevent the UAV from flying in straight lines a square obstacle of 1 m is placed in the middle of the flight arena. A reference velocity of 1 m/s is selected, which results in a turning radius of 0.5 m. This turning radius is used to calculate the collision states. The environments with the smallest and highest collision state percentage are shown in Figure 18. A snapshot and top-view of a real-flight is shown in Figure 19.

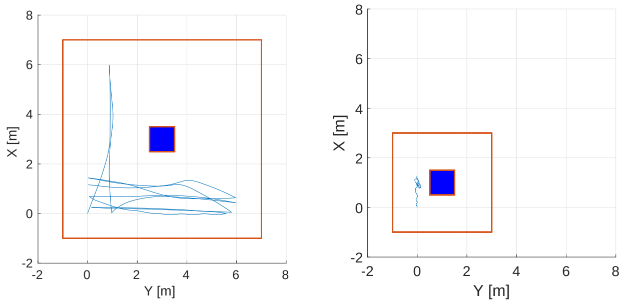


Fig. 18: Environments with the highest and lowest collision state percentage

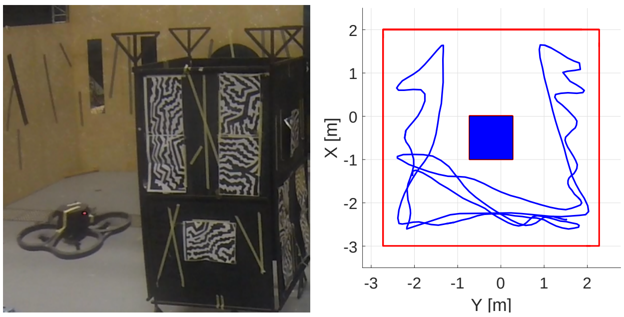
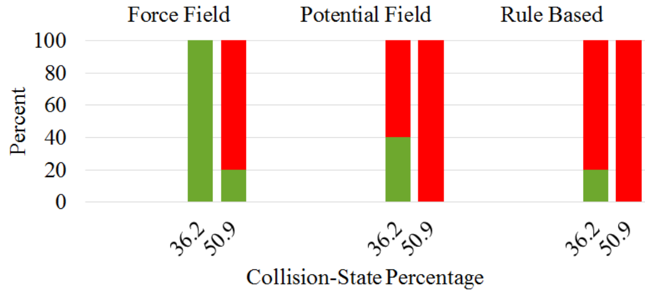


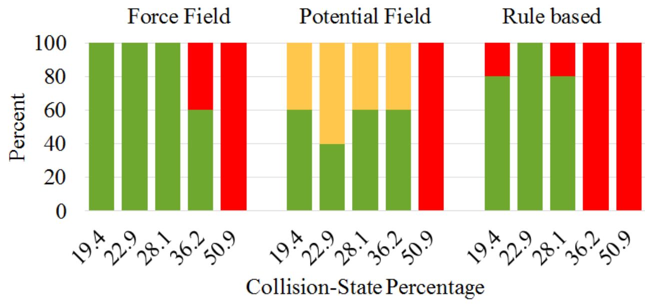
Fig. 19: Snapshot (left) and plotted top-view (right), of a collision state factor performance test.

Again the percentages of the three flight scenarios are evaluated (Figure 20). Due to practical constraints only two collision state factors are shown for real-flight tests. The force field method performed the best. It was able to perform successful flights for a collision state percentage of 36.2. The potential field and rule based method are less successful and not able to avoid obstacles at this percentage.

The simulation results show that the potential field method gets stuck in local minima for percentages lower than 50.9. This is due to the corners in the room, from which the method was sometimes not able to escape. The worst performance is shown by the rule based method. For the real-flights and the simulation the rule based method has a high collision probability for collision state percentage of 36.2 or higher.



(a) Real-flight tests

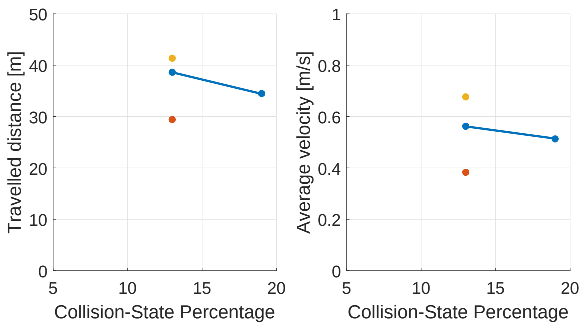


(b) Simulations

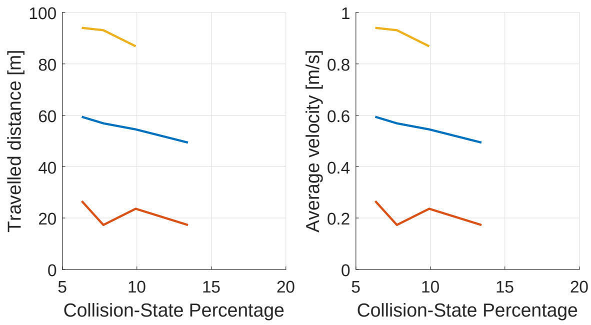
Fig. 20: Percentage of flights that reach the goal (green), do not reach the goal but do not collide (orange) or result in a collision (red).

When looking at the path optimality shown in Figure 21, it can be seen that the plots of the travelled distance and the average velocity are very similar. This is caused by the fixed flight time of the tests. For the velocity the same hierarchy between the three methods can be observed as the one shown in Figure 17. For the simulations a drastic performance difference is present between the three methods. The average velocity of the rule based method is almost twice the average velocity of the force field method and more than four times the velocity of the potential field method. For the real flight tests the difference between the three methods is still there, but smaller.

Because of this difference, the conclusion that the rule based method is the worst, should be questioned. It shows a higher collision probability but it travels further and faster. A second observation is that the average velocity decreases as the collision state factor increases. This is caused by the increased amount of manoeuvres, which is required at higher collision state factors. A manoeuvre always slows down the system so a lower average velocity is found.



(a) Real-flight tests



(b) Simulations

Fig. 21: Path optimality for force field (blue), potential field (red) and rule based (yellow) method.

C. *Average avoidance length*

The effect of the length of the avoidance manoeuvre is measured by increasing the width of a single obstacle in front of the UAV. The smallest obstacle has a width of 0.3 m, the largest a width of 4 m, which are shown in Figure 22.

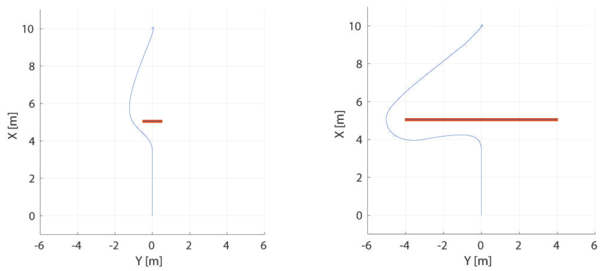
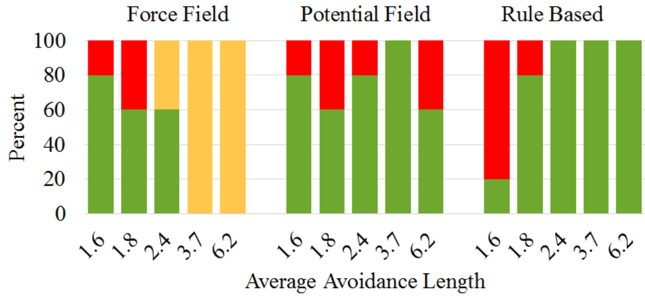


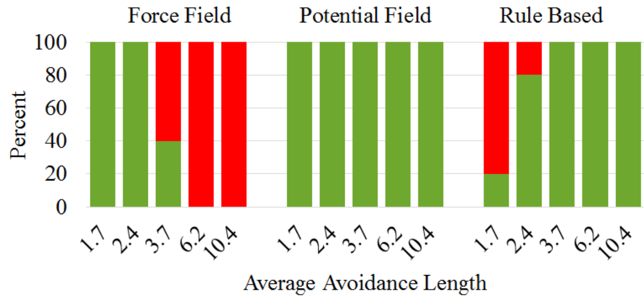
Fig. 22: Simulation environment with an obstacle width of 1 m (left) and 8 m (right).

The collision percentages are shown in Figure 23. The force field method is not able to avoid the obstacle for a value of 3.7 or higher. Instead it gets stuck into a local minimum. The potential field method is able to avoid obstacles for all values. The rule based method is able to avoid large obstacles but unable to avoid small ones. A similar behaviour is observed for the simulations

The path optimality for the different avoidance lengths is shown in Figure 24. Similar travelled distances can be

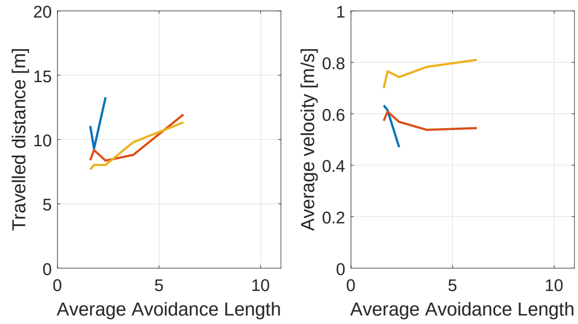


(a) Real-flight tests

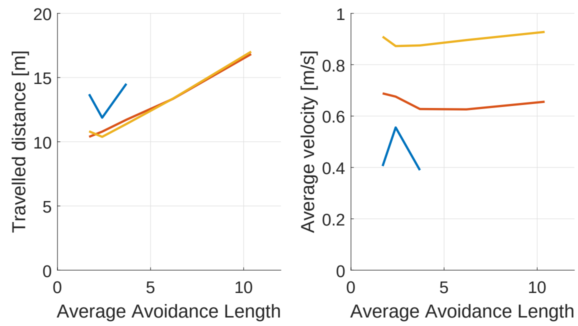


(b) Simulations

Fig. 23: Probability of Flights that reach the goal(green), do not reach the goal but do not collide(orange) or result in a collision(red).



(a) Real-flight tests



(b) Simulations

Fig. 24: Path optimality for force field (blue), potential field (red) and rule based (yellow) method.

observed for the potential and rule based method. The force field method is less optimal showing larger distances. This is caused by the oscillatory motion which is typical to this method. From the average velocities it can again be concluded that the rule based method is the fastest method.

D. Dead-End percentage

The fourth metric is the dead-end percentage. To increase the dead-end percentage one could increase the dead-end area of an obstacle or increase the number of obstacles. In this paper the second option is chosen. The simulation environment with the lowest and highest dead-end factor are shown in Figure 25.

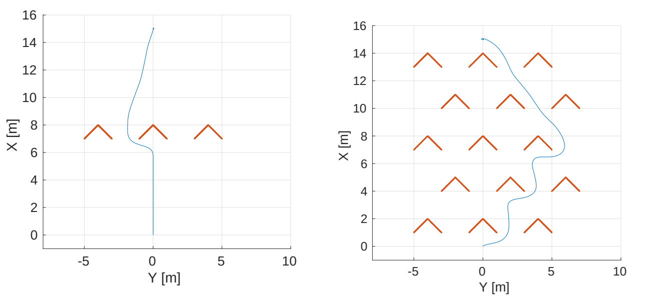


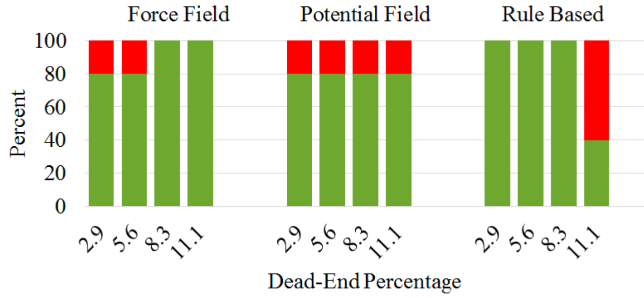
Fig. 25: Environments with the lowest (left) and highest (right) dead-end percentage.

The results of the flight-tests and simulations can be seen in Figure 26. The first thing that stands out is the major difference between the simulation results and the flight test results. For real tests all three strategies are able to reach the goal but in simulation the force field method and the potential field method are not. The force field method crashes for all dead-end percentages, while the potential field method gets stuck in local minima. For both measurements (real-flight and simulation) no influence of the dead-end percentage on the collision probability can be seen.

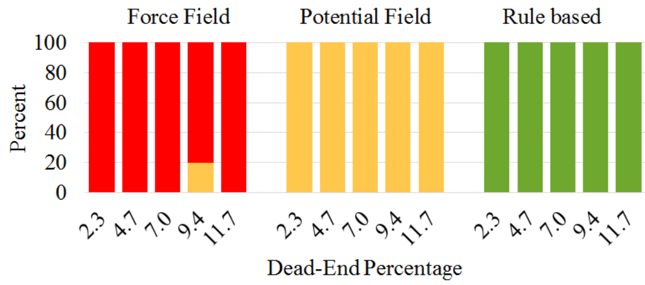
The reason for this result is a combination of the non-homogeneous textures in real-flight and the specific positioning of the obstacles. The non-homogeneous textures forces the methods to turn into one direction. A similar effect occurs when obstacles are not symmetrically aligned with respect to the flight-path. This effect can be mitigated by using larger obstacles. More coherent results are expected when the dead-end areas for each obstacle are increased.

The optimality of the paths is shown in Figure 27. A large drop is observed for all methods at a high dead-end percentage. This occurs due to the positions in which the obstacles where placed. Due to space constraints the obstacles had to be placed in a configuration for which no large avoidance manoeuvre was required.

This points out the difficulty of this metric. The position and orientation of the obstacle highly influence the performance outcome but do not influence the dead-end factor. Such effect is also present in other metrics but to a lesser extend. This can

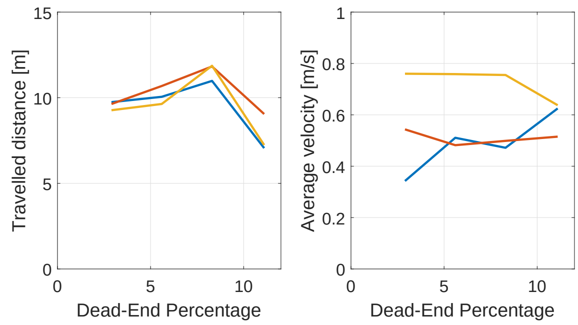


(a) Real-flight tests

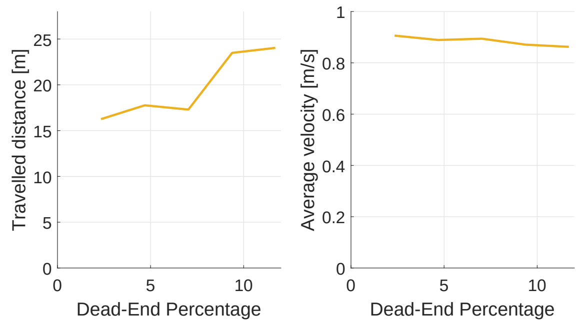


(b) Simulations

Fig. 26: Percentage of flights that reach the goal (green), do not reach the goal but do not collide (orange) or result in a collision (red).



(a) Real-flight tests



(b) Simulations

Fig. 27: Path optimality for force field (blue), potential field (red) and rule based (yellow) method.

be solved by performing more flights, with different initial and goal states, or by flying randomly inside a room as was done for the collision state percentage.

E. Average orientation angle

The final metric is the orientation of the obstacle, which is quantified as the average angle between the flight-path and the obstacle. Again five different values are tested. Two can be seen in Figure 28.

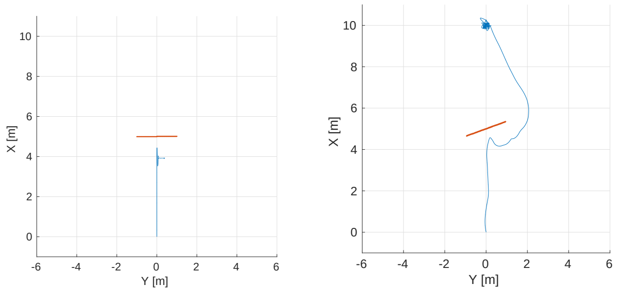


Fig. 28: Simulation environment with an angle of $1°$ (left) and $20°$ (right).

One would expect better performance at higher angles. This effect is indeed seen in the collision probabilities for the force field method shown in Figure 29a. For an angle of $1°$ a high probability is present for the UAV to get trapped in a local minimum. For the potential and rule based method no influence of the obstacle orientation is seen.

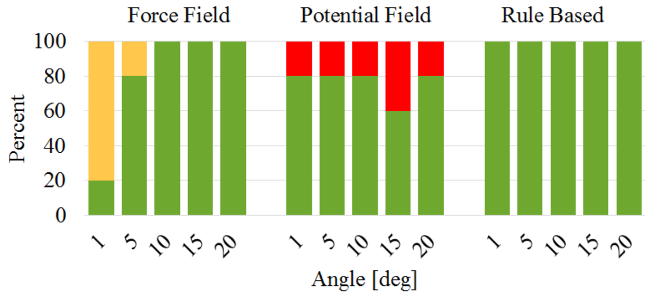
When looking at the optimality of the successful paths it can be concluded that the obstacle orientation does not have a large effect on the travelled distance. It does have an effect on the average velocity.

This effect is mostly visible for the force field method, which shows an increase in average velocity. The inclined obstacle 'pushes' the UAV in a certain direction, resulting in a higher average velocity. Also a small velocity increase can be seen for the potential field method. A smaller heading change is needed to avoid the obstacle and therefore a higher velocity can be maintained. No significant effect for the rule based methods can be seen. Similar conclusion can be drawn from the simulation results shown in Figure 30.

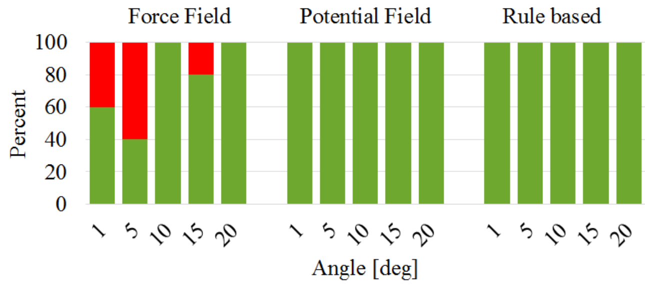
XI. Conclusion evaluation

The shown detection and avoidance tests provide a quantitative analysis of an obstacle avoidance system. Such analysis is new in the field of obstacle avoidance and can be used to define the operational conditions in which an obstacle avoidance system can safely fly. The results also provide a comparison between three avoidance methods. The choice for which method to use depends on the design requirements.

For detection the found performance limits can be summarised as follows: distance < 4 m, illumiance > 7 lx, texture > 20 8-bit grayscale and a texture angle $> 30°$. For the avoidance tests the results are summarised in Table II.

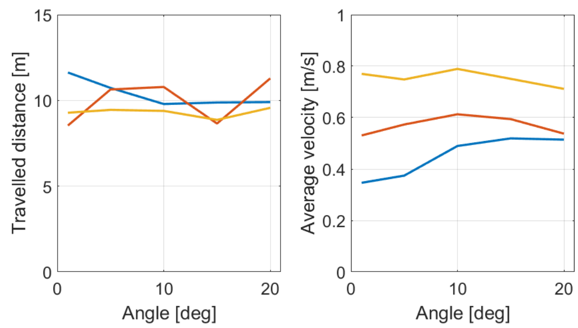


(a) Real-flight tests

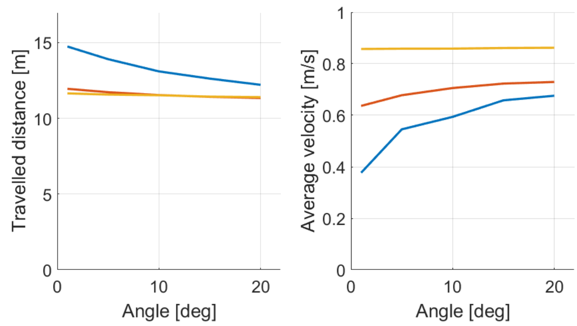


(b) Simulations

Fig. 29: Percentage of flights that reach the goal (green), do not reach the goal but do not collide (orange) or result in a collision (red).



(a) Real-flight tests



(b) Simulations

Fig. 30: Path optimality for force field (blue), potential field (red) and rule based (yellow) method.

TABLE II: Performance summary

	Force Field	Potential Field	Rule Based
Traversability	≥ 6.1	≥ 6.1	≥ 8.6
Collision state percentage	≤ 36.2	≤ 28.1	≤ 28.1
Avoidance length	≤ 2.4	ALL	≥ 1.8
Dead-end percentage	= 0	= 0	ALL
Orientation angle	≥ 5	ALL	ALL

The table shows the conditions under which the algorithm is able to avoid obstacles. The force field method is able to avoid obstacles with a higher traversability of 3.9 for example. When a method was able to avoid obstacles for the complete range of values it is specified as 'ALL'. If the method was not able to avoid any obstacle at all it is specified as '=0'. Such table could be used by engineers to develop obstacle avoidance systems which can operate under a specific set of operational conditions.

For example for an environment which consist of a large open space with sparse poles, the rule based method would not be the best method. In Table II it can be seen that this method is not able to avoid obstacles with a smaller average avoidance length of 1.8. The potential field method would be a better choice, since it is able to avoid obstacles with a small average avoidance length. A different choice is made when the restaurant does not contain poles but small walls. In such case the rule based method is appealing, since it is able to avoid such obstacles faster than the potential field method.

XII. Evaluation of the Framework

From the measurements a general idea is created in which the proposed application could operate. A quantified limit is found which can be used to predict the performance of an obstacle avoidance system in new environments. The downside of the framework is that the tests do not provide a complete 'avoidance envelope'. In the evaluation framework only one environment metric is changed to see the effect on the performance. In real flight these environments happen simultaneous. So extra tests are needed to determine the performance degradation when environment metrics are changed simultaneously. To do this the system should be tested in combinations of environment metrics, but this would require many tests. The number of tests grows exponential with the amount of metrics, it is therefore crucial to only select the relevant combinations of metrics.

By adding relevant combinations to the framework the evaluation method could be improved. Such relations have already been presented in this paper. With the relation between the contrast and the illumination shown in Figure 10 but also with the relation between distance and the texture angle (Figure 11). It will be difficult to know all dependencies in advance, but by using a structured evaluation method such knowledge can slowly grow.

Another consecutive step is to do a robustness check of the environment metrics. The collision percentage for example could be influenced in different ways. It should be checked if the performance is sensitive to these differences. Also the sensitivity with respect to the initial and goal state should be checked. This could be done by randomizing the initial and goal positions as suggested in subsection X-D.

A final suggestion to the framework is to also look at other independent variables besides the environment metrics. For example the tuning variables of an avoidance method or the amount of detection noise.

XIII. Conclusion

A new framework was proposed, which allows the quantification of the strengths and weaknesses of an obstacle avoidance system. The framework identifies key obstacle avoidance functions in the control loop and introduces novel performance and environment metrics to quantify the performance of these functions. The application of the framework to a specific UAV 2D avoidance task shows that the metrics allow to identify the limits of the avoidance system in an objective and quantifiable manner. In this sense, the framework hopefully forms an important step towards a more solid design, evaluation, and comparison of obstacle avoidance methods for robotics.

References

[1] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," vol. 57, pp. 65–100, 2010.

[2] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert, "Learning monocular reactive UAV control in cluttered natural environments," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1765–1772, 2013.

[3] D. Magree, J. G. Mooney, and E. N. Johnson, "Monocular visual mapping for obstacle avoidance on UAVs," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 74, pp. 17–26, 2014.

[4] A. Bachrach, R. He, and N. Roy, "Autonomous Flight in Unknown Indoor Environments," *International Journal of Micro Air Vehicles*, vol. 1, no. 4, pp. 217–228, 2010.

[5] S. Hrabar, "3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 807–814, 2008.

[6] B. Mettler, Z. Kong, C. Goerzen, and M. Whalley, "Benchmarking of obstacle field navigation algorithms for autonomous helicopters," *Proceedings of the 66th Annual Forum of the American Helicopter Society*, pp. 1–18, 2010.

[7] J. K. Kuchar and J. K. Kuchar, "Safety Analysis Methodology for UAV Collision Avoidance Systems," *Regulation*, 2005.

[8] A. Alexopoulos, A. Kandil, P. Orzechowski, and E. Badreddin, "A Comparative Study of Collision Avoidance Techniques for Unmanned Aerial Vehicles," *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1969–1974, 2013.

[9] P. Viswanathan, J. Boger, J. Hoey, and A. Mihailids, "A Comparison of Stereovision and Infrared as Sensors for an Anti-Collision Powered Wheelchair for Older Adults with Cognitive Impairments," *2nd International Conference on Technology and Aging (ICTA)*, 2007.

[10] N. a. Thacker, A. F. Clark, J. L. Barron, J. Ross Beveridge, P. Courtney, W. R. Crum, V. Ramesh, and C. Clark, "Performance characterization in computer vision: A guide to best practices," *Computer Vision and Image Understanding*, vol. 109, no. 3, pp. 305–334, 2008.

[11] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, 2000.

[12] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of Intelligent and Robotic Systems*, 2008.

[13] B. M. Albaker and N. a. Rahim, "Unmanned aircraft collision detection and resolution: Concept and survey," *Proceedings of the 2010 5th IEEE Conference on Industrial Electronics and Applications*, pp. 248–253, 2010.

[14] A. Mujumdar and R. Padhi, "Evolving Philosophies on Autonomous Obstacle/Collision Avoidance of Unmanned Aerial Vehicles," *Journal of Aerospace Computing, Information, and Communication*, vol. 8, no. February, pp. 17–41, 2011.

[15] K. Sebesta and J. Baillieul, "Animal-inspired agile flight using optical flow sensing," *Proceedings of the IEEE Conference on Decision and Control*, no. 1, pp. 3727–3734, 2012.

[16] S. Karaman and E. Frazzoli, "High-speed flight in an ergodic forest," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2899–2906, 2012.

[17] C. D. Wagter, S. Tijmons, B. D. W. Remes, and G. C. H. E. D. Croon, "Autonomous Flight of a 20-gram Flapping Wing MAV with a 4-gram Onboard Stereo Vision System," *IEEE International Conference on Robotics & Automation (ICRA)*, pp. 4982–4987, 2014.

[18] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[19] M. Kytö, M. Nuutinen, and P. Oittinen, "Method for measuring stereo camera depth accuracy based on stereoscopic vision," pp. 78 640I–78 640I–9, 2011.

[20] A. a. Kandil, A. Wagner, A. Gotta, and E. Badreddin, "Collision avoidance in a recursive nested behaviour control structure for unmanned aerial vehicles," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, pp. 4276–4281, 2010.

[21] W. H. Huang, B. R. Fajen, J. R. Fink, and W. H. Warren, "Visual navigation and obstacle avoidance using a steering potential function," *Robotics and Autonomous Systems*, vol. 54, pp. 288–299, 2006.

[22] E. J. J. Smeur, "Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Aerial Vehicles," no. February 2016, pp. 1–16, 2015.

Part II

Conference paper

Performance Evaluation in Obstacle Avoidance

Clint Nous, Roland Meertens, Christophe de Wagter and Guido de Croon
Delft University of Technology

***Abstract*—No quantitative procedure currently exists to evaluate the obstacle avoidance capabilities of robotic applications. Such an evaluation method is needed for comparing different methods, but also to determine the operational limits of autonomous systems. This work proposes an evaluation framework which can find such limits. The framework comprises two sets of tests: detection tests and avoidance tests. For each set, environment and performance metrics need to be defined. For detection tests such metrics are well known. For avoidance tests however such metrics are not readily available. Therefore a new set of metrics is proposed. The framework is applied to a UAV that uses stereo vision to detect obstacles. Three different avoidance methods are compared in environments of varying difficulty.**

***Index Terms*—Obstacle Avoidance, Evaluation Framework, Complexity Metrics, Benchmark**

I. Introduction

Autonomous flight with aerial robots has many promising applications, such as surveillance, inspection or package delivery. To carry out such tasks a reliable obstacle avoidance system is essential. Many obstacle avoidance systems are available [1], but it is unknown what the performance of these systems is. No quantitative evaluation procedure to measure the performance of obstacle avoidance systems is available.

Such a procedure is required to determine the reliability of obstacle avoidance systems and to find in what conditions these systems can safely operate. Knowledge of these operational conditions is crucial when deploying UAVs for practical applications.

The functioning of an obstacle avoidance system is often demonstrated in a single environment. Environments found in literature are divers, with amongst others: forests, buildings, hallways and sparse obstacle courses. It is difficult to make performance predictions based on a single environment since the performance of an obstacle avoidance system depends on the environment in which it operates. For the same reason it is difficult to compare obstacle avoidance methods which are demonstrated in different environments.

Therefore a standard evaluation procedure is needed. Such procedure would: (1)provide a quantitative measure of the system performance, (2)assist in designing engineering solutions, (3)compare obstacle avoidance methods and (4)allow accurate assessment of the state of the art. Without a good evaluation method the development of new algorithms is likely to lead to ad-hoc solutions, which is currently seen in the field of obstacle avoidance.

An attempt to create such an evaluation method is proposed by Mettler et al. [2]. In this method six simple obstacle courses and an urban environment are used to test the performance

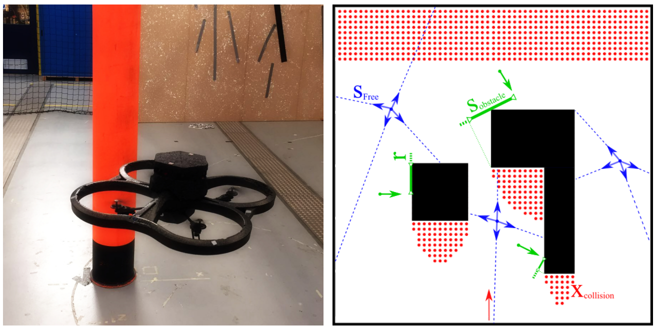


Fig. 1: The framework is applied for an ARDrone with an omnidirectional stereo vision system (left). In the framework environment complexity metrics are introduced, which provide a guideline for tests in different environment conditions (right).

of an obstacle avoidance algorithm. Unfortunately Mettler does not motivate the choice for the selected tests and no explanation is given on how representative these tests are for the overall performance of a system.

Besides this work and some papers in which obstacle avoidance algorithms are compared [3], researchers have not attempted to quantify or benchmark the performance of obstacle avoidance algorithms. This is remarkable, considering the countless research contributions done in the field. Such evaluation methods and benchmark data sets are common practice in other research fields such as computer vision or control engineering.

In this paper an evaluation framework is proposed which makes it possible to quantify the performance of an obstacle avoidance system. A key aspect of this framework is to quantify the environment using specific metrics. First the evaluation framework and its metrics are discussed (section II & III) after which the framework is applied to a robotic application (section IV).

II. Evaluation Framework

Developing an evaluation framework for obstacle avoidance is difficult due to three aspects: 1. Performance depends on the complete control loop, 2. There is a high variety of operational conditions, 3. Each obstacle avoidance method is developed for a different set of environments.

The first difficulty is caused by the relation between the obstacle avoidance system and the platform on which it is applied. The performance is dependent on this relation. Therefore no direct comparison can be made between obstacle avoidance methods applied on different platforms. The second and third

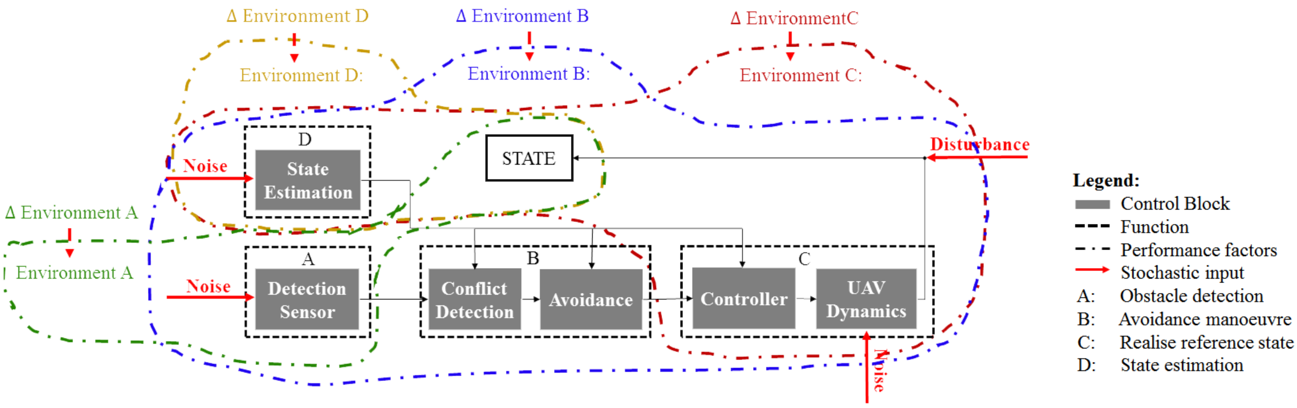


Fig. 2: Obstacle avoidance control loop in which the performance dependencies of four functions(A-D) are visualised.

challenges are caused by the performance dependency on the environment. These environment conditions are diverse and sometimes difficult to quantify. Since each method can be sensitive to different environment conditions no 'one size fits all' test setup is possible. Different tests are needed for different methods. In this section, we analyse the entire obstacle avoidance control loop and identify what parts of the loop can be evaluated separately, and what factors will influence the evaluation.

The complete control loop of an obstacle avoidance system is shown in Figure 2. In the figure four main functions are identified (A-D), which are shown by dotted squares. For each function a 'cloud' is drawn to visualise the factors that impact the performance of each function. So it specifies in which way the performance depends on the control loop.

This paper focuses on the obstacle detection function(A) and the calculation of the avoidance manoeuvre(B), these are the primary functions of an obstacle avoidance system. The factors which influence the performance of the detection function are visualised by the green cloud. It can be seen that three factors are present: state, noise and the environment. The state determines the distance to an obstacles but also the velocity. Both influence detection performance. The noise arrow represents the internal noise of the sensor and has a direct impact on the measurement. Finally the environment conditions to which the detection sensor is sensitive is represented by 'Environment A'.

The factors that influence the performance of function B is shown by the blue 'cloud'. All blocks in the control loop effect this performance. Also noise, disturbances and external conditions are included. Lastly the environment for which the performance of the avoidance manoeuvre is sensitive is included and represented as Environment B. For completeness also the 'clouds' of function C and D are drawn.

Now an overview is presented of what affects the performance of obstacle avoidance functions, it can be used to define evaluation tests. Each factor in the clouds of function A and B can be used as an independent variable in a performance tests. In this paper the focus is put on the environment factor. The others factors are assumed to be constant. These constants should be explained since the performance is still dependent on them.

To quantify the environment factor, (such that it can be used as an independent variable) it needs to be described as a metric. When the environment is not described using specific metrics it becomes difficult to make performance predictions. The definition of these metrics will be discussed in the next section. Besides the independent variable a dependent variable is needed to quantify the performance. So far it has not been discussed how performance is defined. This can be done by specifying a performance metric. Both metrics are discussed for function A (detection) and function B (avoidance) in the next section.

III. Environment and Performance Metrics

The environment and performance metrics selected for the evaluation measurements depend on the obstacle avoidance system that is used. For the detection function these metrics depend on the sensor and its processing, for the avoidance function these metrics depend on the avoidance algorithm.

Also some general characteristics influence the type of metric. For instance, an obstacle avoidance task with static obstacles requires different environment metrics than an obstacle avoidance task with dynamic obstacles. In the dynamic case, additional factors such as the velocities and reactions of the obstacles play an important role and should be captured by the environment metrics. The characteristic whether the task has the UAV move in 2D or 3D has a similar influence on the metrics, as 2D environment metrics may not directly generalise to the 3D case. Although the framework is applicable to different such characteristics, here the focus will be on metrics for a 2D environment with static obstacles.

A. Detection

The goal of the detection measurement is to determine under what conditions obstacles can be detected. To define metrics for these tests a broad overview of detection methods is required. The main methods can be divided based on the detection sensor that is used. The six main sensors are: monocular-vision, stereo-vision, infrared, sonar, laser and radar. In this

TABLE I: Relevant environment metrics for detection sensors

	Monocular vision	Stereo Vision	Infrared	Sonar	Laser	Radar
Distance	✓	✓	✓	✓	✓	✓
Reflectivity	✓	✓	✓	✓	✓	
Illumination	✓	✓				
Texture	✓	✓				
Illumination (Infrared)			✓			
Inclination			✓	✓	✓	
Transparency	✓	✓			✓	
Radar cross section						✓

discussion cooperative sensors such as ADS-B are omitted. First the environment metrics are discussed.

1) Environment metric: Each sensor is sensitive to different environment characteristics. Fortunately these are fairly well known. An overview of these metrics can be seen in Table I. In the table the relevant metrics for each sensor is specified. The distance, for example, is relevant for all sensors but the illumination is only relevant for monocular and stereo vision.

2) Performance metric: Performance metrics for detection seen in literature are the distance error and variance. Another metric is the Receiver Operating Characteristic (ROC) curve in which true positives and false positives are plotted as function of a threshold. A third metric seen in literature is the computational time. Since the computational time does not depend on the environment this metric is not seen as a performance metric in this framework but rather as a condition.

B. Avoidance

The goal of the avoidance tests is to determine under which conditions detected obstacles can be avoided. Again relevant metrics need to be selected which are dependent on the method that is used. Unfortunately no simple division can be made between the wide diversity of avoidance methods. Methods vary from simple rule based instructions to complex path planners. Even within path planning a large variety exist. All these methods can be sensitive to different environment conditions. A similar table as Table I needs to be constructed in which the columns represent the methods and the rows the complexity metrics. But is unknown what these metrics should be. In the following a novel set of such metrics is proposed.

1) Environment metric: For detection sensors the relation between the environment and the performance is often discussed in literature, but for avoidance algorithms it is not. As mentioned in the introduction, most research contributions use a specific environment without a motivation or specific metric. Therefore not much is known about the performance of these algorithms in different environments.

Only a few environment metrics are seen in literature: the width of the obstacles, in-between distance or the density [4],[5]. These metrics do not take the size of the UAV into account, while this is essential. It is more challenging for a UAV with a radius of 0.5 m to fly trough obstacles with an in-between distance of 1.0 m, than it is for a UAV with a radius of 0.1 m. In the following a new set of non-dimensional environment metrics is proposed which take the properties of the UAV into account.

a) Traversability: The fist metric is the traversability which is related to the obstacle density. An obstacle avoidance task becomes more difficult when the distance between obstacles becomes smaller. This difficulty is quantified by the traversability. The traversability is calculated by selecting random positions and headings in a flight space. For these positions and headings the maximum straight-flight distance s is determined. The average of these distances over n samples gives a measure of how densely packed the environment is and therefore how challenging the performance task is. In Figure 1 the blue lines display these distance for four points and headings. This calculation is shown by Equation 1. In Equation 1 the value is divided through the radius r of the UAV to make the metric non-dimensional.

$$Traversability = \frac{1}{n \cdot r} \sum_{0}^{n} s \tag{1}$$

b) Collision state percentage: The second metric is the collision state factor, which combines the dynamical constraints of the UAV with the available flight space. Flying inside a room with obstacles becomes more difficult when the turning radius of the UAV increases, but also when the size of the room decreases. This effect is quantified by calculating the percentage of states for which a collision is unavoidable. To determine these states each state is propagated into the future. When all propagated trajectories lead to a collision, the state is marked as a 'collision state'. For $\psi = 0$ these are shown by the red dotes in Figure 1. The propagated trajectories are calculated using a minimum turning radius which depends on the UAV dynamics, the velocity and the delay in the system.

c) Average avoidance length: The third proposed metric is similar to the size of the obstacle. This metric quantifies the difference between a forest environment with thin obstacles and a building environment with wide obstacles. It is calculated by averaging the needed lateral movement to avoid obstacles at each time-step during a flight. The lateral movement is sum of the (projected) width of the obstacle and the radius of the UAV, as shown by the green lines in Figure 1. Again the metric is made non-dimensional by dividing it through the radius of the UAV.

d) Other metrics: Two other metrics, which are not discussed in detail, are the average orientation of obstacles and the percentage of dead-ends in a flight space. These have been inspired by known weak points in force field methods and path planners. This set of five metrics could be expanded further by introducing more novel metrics.

2) Performance metrics: The performance metrics are again fairly straightforward. Generally three types of performance metrics are seen: computational time, success rate and path optimality. There exists a hierarchy between success rate and path optimality, since path optimality can only be determined when an obstacle is successfully avoided.

For the success rate three scenarios are distinguished: successful flights which reach their goal safely, unsuccessful flights which do not reach the goal but do not collide either, and flights which lead to a collision. The performance metric specifies the percentage for each scenario.

Two secondary performance metrics evaluate the optimality of the successful flights. This can be done in multiple ways. In this framework the choice is made to focus on two optimality metrics: travelled distance and average velocity. The duration and energy usage can be derived from these metrics.

IV. Evaluation of a robotic application

In this section the framework is applied to a robotic application which can be seen in Figure 6. The shown UAV makes use of six stereo cameras to detect obstacles and has a choice between three different avoidance strategies. In the following subsections the performance of this obstacle avoidance system is determined.

A. Detection Performance

The six 4-gram stereo cameras [6] create a 360° view of the environment. The cameras produce two small images of 96×128 px for which pixels are sparsely matched on the stereo board to create a disparity map. This map is created using the Sum-of-Absolute-Difference scheme presented in [7]. For stereo vision five performance tests are suggested in Table I. Three of these tests are presented in the following paragraphs.

1) Distance: For the first test the distance is selected as independent variable. The performance effect of the obstacle distance is measured by placing an obstacle of 1×1 m (Figure 3) in front of the stereo-camera. The distance to this obstacle is varied from 0.5 m to 4 m. The result can be seen in Figure 3. The figure shows that an obstacle can be measured accurately

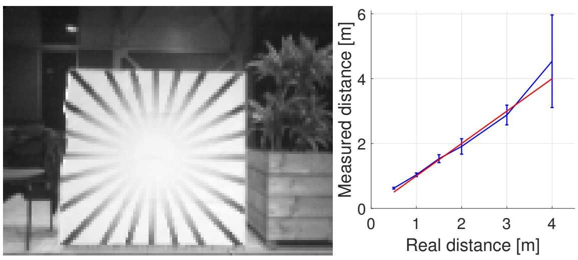


Fig. 3: Test obstacle (left); Results distance test, red = ground truth, blue = stereo measurements (right).

up to three meters. For larger distances the standard deviation increases rapidly. At four meters a standard deviation of more than 1 m is present. This increase in variance is fundamental to stereo vision. So for this detection sensor a range of 3 m can be assumed.

2) Illumination: The second selected environment metric is the illumination. Measurements were conducted in a theatre in which the light exposure could be controlled. Tests in an illumination regime from 284 lx to 0.05 lx were conducted. The results of these measurements are shown in Figure 4. A decrease in illuminance results in a small increase in detection error, but it remains remarkably accurate for distances of 0.5 to 3 meters. The stereo camera is still able to detect obstacles up to an illumination of $10^{-0.7}$ lx.

A different behaviour is seen at a distance of 4 m. At an illumination of $10^{0.85}$ lx the distance estimate increases up to a value of 9 m. When the illuminance is lowered further the distant measurement drops to a value of 0.4 m. The reason of this large underestimation is caused by different illumination of the left and right camera. So not only the illuminance is critical in detecting obstacles but also the distribution of light. According to Figure 4 a minimal illuminance of 7 lx is required to obtain results at which the different distances can be well discerned.

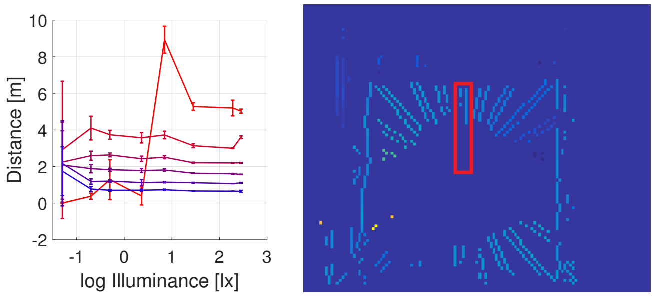


Fig. 4: Results illuminance test, distance of 0.5 = blue to 4 m = red (left); Disparity map from test obstacle at 2 m (right).

3) Texture contrast: The third metric which is analysed is texture contrast. One 'ray' in Figure 3 is analysed. In each ray the grey-scale is decreased from white in the centre to black at the edges. By doing this the contrast increases from zero to a maximum between black and white. At a certain contrast the stereo algorithm is able to find a match, which is shown in Figure 4. The distance measurement of the matches found inside the red area are shown in Figure 5. The figure

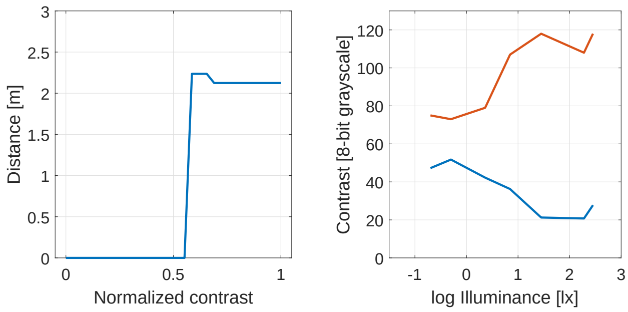


Fig. 5: Results contrast test (left); Contrast vs illuminance test, blue = required contrast, red = maximum contrast (right).

shows that the contrast does not influence the accuracy of the measurement but rather the point at which a detection is possible. Since a pixel difference depends on the contrast of the obstacle and the illuminance, it would be interesting to see how this switching point changes when the illuminance is decreased.

This relation is shown in Figure 5. The red line in the figure shows the difference in 8-bit gray-scale at the edge of the panel. The contrast decreases when the illumination is decreased. The blue line shows the 8-bit gray-scale value at which a first match is found. A minimum contrast of 20 8-bit gray-scale is required to find a match. This value increases when the illuminance is decreased. This effect can be explained by the increased noise in the image. Since a point is only accepted when it 'stand outs' from the other pixels, a higher value is needed to find a disparity.

B. Avoidance

Now that an idea of the detection performance is present, the performance of the avoidance manoeuvre can be analysed. Three avoidance strategies are applied: A force field method based on the work from Kandil et al. [8], a potential field method based on the work from Huang et al. [9] and a simple rule based method. This method selects the heading closest to the goal heading in which no obstacle is present[1].

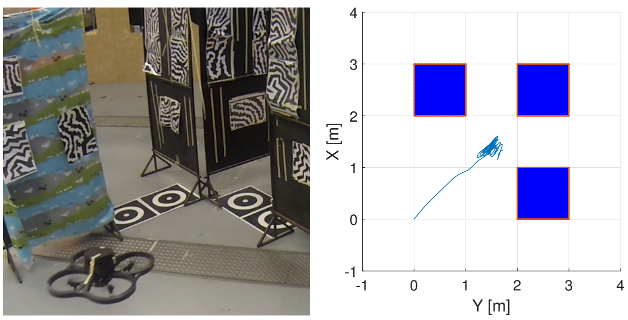


Fig. 6: Snapshot (left) and plotted top-view (right), of a traversability performance test

For these methods it is not clear which metrics are relevant. Therefore all proposed metrics in section III are used for the performance test. The measurements can be performed in a simulation or in real-flight. For both measurements it is important to state the assumptions made in the complete control loop as described in section II. Here the results of real-flight tests are presented[2]. The measurements are performed on an ARDrone2.0 using the paparazzi autopilot. An INDI attitude controller [10] is used combined with a PID velocity controller. The states are estimated using an Optitrack system. Detection noise is estimated to have a standard deviation of 0.1 m, state noise, actuator noise and disturbances are assumed to be zero. In the following the performance is discussed for three environment metrics: traversability, collision state factor and average avoidance length.

1) Traversability: The performance under different traversability values is tested by decreasing the distance between three obstacles. The obstacles consist of 1 m square blocks and are changed from an in-between distance of 1 m up to 3 m. The test with the lowest traversability factor is shown in Figure 6.

[1] github.com/paparazzi/tree/master/sw/airborne/modules/obstacle_avoidance
[2] youtube.com/playlist?list=PL_KSX9GOn2P8DI_viJWg_hpG-uBLP-yNK

A total of five tests were performed. Each test consisted of five flights. The results of these tests can be seen in Figure 7 and 8. Figure 7 shows the percentages of the three flight scenarios discussed in section III.

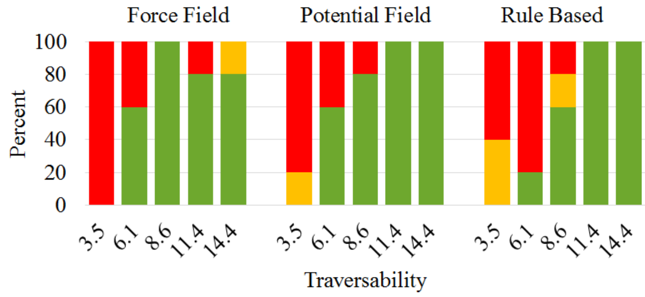


Fig. 7: Percentage of runs that reach the goal (green), not reach the goal but no collision (orange) or result in a collision (red).

It can be seen that all three methods are not able the reach their goal for a traversability of 3.5. For a value of 6.1 and higher the majority of flights for the force and potential field method are successful. The rule based method however still has a high collision rate at this value. This higher collision percentage is likely the cost for the relative high velocity. This velocity difference can be seen in Figure 8. The figures shows the path optimality of the successful flights for each method.

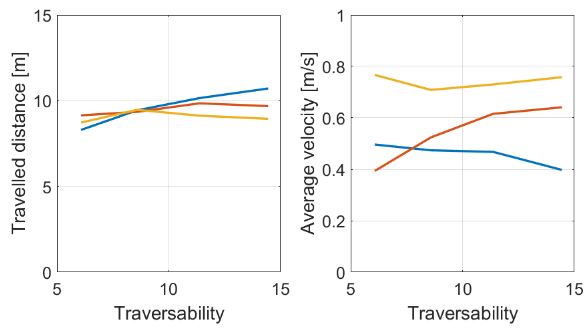


Fig. 8: Path optimality vs traversability; Force field (blue), Potential field (Red), Rule based (yellow).

2) Collision state percentage: The performance under different collision state factors is measured by decreasing the room in which it flies. To prevent the UAV from only flying in straight lines a square obstacle of 1 m is placed in the middle of the flight arena. A reference velocity of 1 m/s is used. An example of a test-flight is shown in Figure 9. Again the percentages of the three flight scenarios are evaluated, as well as the path optimality. The force field method performed the best, it was able to perform successful flights up to a collision state percentage of 19%. The potential field and rule based method are less successful and able to avoid obstacles up to a state factor of 13%. For path optimality the same hierarchy was present as the one shown in Figure 8. Also a decreased average velocity for all methods could be observed when the collision state factor was increased.

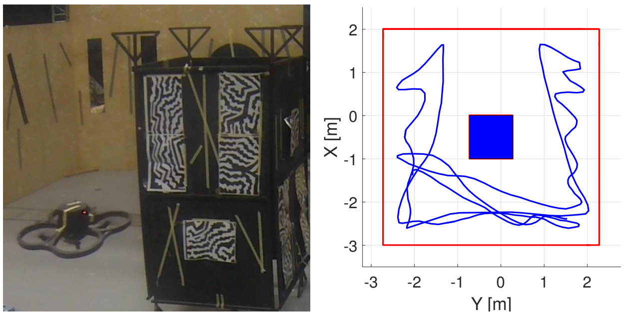


Fig. 9: Snapshot (left) and plotted top-view (right), of a collision state factor performance test

3) Average avoidance length: The effect of the length of the avoidance manoeuvre is measured by increasing the width of a single obstacle in front of the UAV. The smallest obstacle has a width of 0.3 m, the largest a width of 4 m. The results of these flights are shown Figure 10.

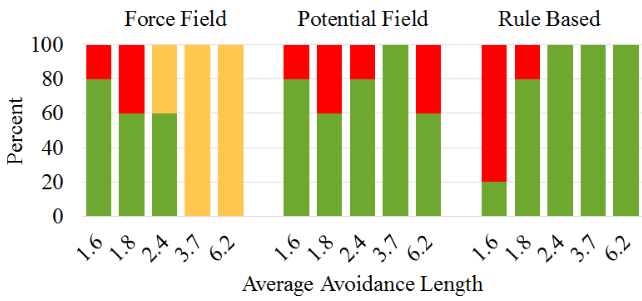


Fig. 10: Percentage of runs that reach the goal (green), not reach the goal but no collision (orange) or result in a collision (red).

Figure 10 shows that the force field method clearly depends on the avoidance length. For a value of 3.7 or higher it is not able to avoid the obstacle. Instead it gets stuck into a local minimum. The potential field method is able to avoid obstacles for all values. The rule based method is able to avoid large obstacles but unable to avoid small ones. Again the path optimality was analysed. Similar travelled distances were observed for the potential and rule based method. The force field methods was less optimal, with larger distances.

C. Conclusion performance tests

The shown test results provide a quantitative analyses of an obstacle avoidance method. Such analysis is new in the field of obstacle avoidance and can be used to define the operational conditions in which an obstacle avoidance system can safely fly. The results also provide a comparison between three avoidance methods. The choice for which method is the best depends on the design requirements.

For the detection the found performance limits can be summarised as follows: Distance < 3 m, Illumiance > 7 lx, and Texture > 20 8-bit grayscale of contrast. For the avoidance test the results are summarised in Table II.

TABLE II: Performance summary

	Force Field	Potential Field	Rule Based
Traversability	>3.9	>3.9	>5.9
Collision state factor	<19	<13	<13
Avoidance length	<2.4	ALL	>1.6
Orientation angle	>1	ALL	ALL
Dead-End factor	= 0	= 0	ALL

The table shows the limits of the proposed methods for each metric. The dead-end factor and average orientation are included as well. Such a table could be used by engineers to design obstacle avoidance systems which can operate under a specific set of operational conditions.

V. Conclusion

A new framework was proposed, which allows the quantification of the strengths and weaknesses of an obstacle avoidance system. The framework identifies parts of the entire obstacle avoidance control loop that can be tested separately, and introduces novel performance and environment metrics. The application of the framework to a specific UAV 2D avoidance task shows that the metrics allow to identify the limits of the avoidance system in an objective and quantifiable manner. In this sense, the framework hopefully forms an important step towards a more solid design, evaluation, and comparison of obstacle avoidance methods for robotics.

References

[1] C. Goerzen, Z. Kong, and B. Mettler, *A survey of motion planning algorithms from the perspective of autonomous UAV guidance*, 2010, vol. 57.

[2] B. Mettler, Z. Kong, C. Goerzen, and M. Whalley, “Benchmarking of obstacle field navigation algorithms for autonomous helicopters,” *Proceedings of the 66th Annual Forum of the American Helicopter Society*, pp. 1–18, 2010.

[3] A. Alexopoulos, A. Kandil, P. Orzechowski, and E. Badreddin, “A Comparative Study of Collision Avoidance Techniques for Unmanned Aerial Vehicles,” *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1969–1974, 2013.

[4] K. Sebesta and J. Baillieul, “Animal-inspired agile flight using optical flow sensing,” *Proceedings of the IEEE Conference on Decision and Control*, no. 1, pp. 3727–3734, 2012.

[5] S. Karaman and E. Frazzoli, “High-speed flight in an ergodic forest,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2899–2906, 2012.

[6] C. D. Wagter, S. Tijmons, B. D. W. Remes, and G. C. H. E. D. Croon, “Autonomous Flight of a 20-gram Flapping Wing MAV with a 4-gram Onboard Stereo Vision System,” *IEEE International Conference on Robotics & Automation (ICRA)*, pp. 4982–4987, 2014.

[7] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[8] A. a. Kandil, A. Wagner, A. Gotta, and E. Badreddin, “Collision avoidance in a recursive nested behaviour control structure for unmanned aerial vehicles,” *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, pp. 4276–4281, 2010.

[9] W. H. Huang, B. R. Fajen, J. R. Fink, and W. H. Warren, “Visual navigation and obstacle avoidance using a steering potential function,” *Robotics and Autonomous Systems*, vol. 54, pp. 288–299, 2006.

[10] E. J. J. Smeur, “Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Aerial Vehicles,” no. February 2016, pp. 1–16, 2015.

Part III

Literature study

Chapter 1

Introduction

Autonomous flight with aerial robots has many promising applications, such as surveillance, inspection or package delivery. To accomplish such tasks a reliable obstacle avoidance system needs to be available. Many systems have been developed (Meier et al., 2012)(Müller et al., 2014), but it is unknown how reliable these systems are and in what conditions these systems can operate. When deploying practical applications these operational conditions need to be known. Unfortunately no method exist to determine these conditions. The goal of this research is to create such a method, which can be applied to determine operational limits for practical applications.

Besides a way to determine the operational limits of practical applications, a standard evaluation method gives other benefits as well. These benefits are set out for standardized evaluation methods in the field of computer vision in (Bowyer & Phillips, 1998), but also apply to standardized evaluation methods in obstacle avoidance:

1. Provide evidence of the workings of the system to potential users.
2. Assist in designing engineering solutions.
3. Allow accurate assessment of the state of the art.
4. Place obstacle avoidance on a solid experimental background.

Or in other words a standard evaluation method allows us the understand the strengths and weaknesses of an obstacle avoidance system, which can be used to determine the focus of research projects. Also comparisons between different methods can be made. From a design perspective the metrics of an evaluation method would make it possible to define requirements for obstacle avoidance systems. These requirements can be compared to those of existing systems and point out where improvements need to be made.

Also in the work form Brady (2006) the merit of a standard evaluation method is stated:

Only with proper and standardized evaluations advances in the field can be identified and promoted.

Brady (2006) also concludes that without a good evaluation method the development of new algorithms are likely to lead to ad-hoc solutions. Something which is currently seen in the field of Obstacle Avoidance (OA). A lot of different algorithms exist but these algorithms all apply to specific cases.

In this study an overview of the literature is given to come up with a standardized evaluation method. The next chapter discusses the challenges in developing such a method and specifies the research questions. In chapter 3 a general overview of the aspects involved in developing an evaluation method are set out. After which detection and avoidance are further elaborated. (chapter 4 and 5) Finally in chapter 6 the conclusions are discussed and the steps for future thesis work are laid out.

Chapter 2

Problem statement and research questions

2-1 Problem statement

In the introduction the advantages of a standardized evaluation method for obstacle avoidance are listed. An attempt to develop such a method is done by Mettler et al. (2010). In his research a standardized method is proposed to evaluate avoidance methods. In this method several obstacle courses in artificial and urban environments are used to test the performance of an obstacle avoidance algorithm. Unfortunately Mettler does not motivate the choice for the selected tests. No explanation is given on how representative these tests are to determine the overall performance of an OA algorithm. Another point of critic is that no assumptions are stated about the way obstacles are detected. It appears Mettler assumes perfect detection, which is not possible in real life situations. A final note on the method of Mettler is that the method does not consider dynamic obstacles, but only takes static objects into account.

Another evaluation method for OA is done in a paper by Alexopoulos et al. (2013), no standardized method is proposed but a comparison between three avoidance methods is being made. The methods are compared based on computational cost and on the amount of collisions in thirteen predefined environments. The method could serve as a standardized evaluation method but has some downsides. Again no explanation is given on how the environments are selected and how representative these are for the overall performance. Also the detection of obstacles is not discussed, which is essential for an OA systems.

The previous two papers give an idea on how a standardized evaluation method for obstacle avoidance could look like, but are both unsuited because of the critics mentioned in the previous paragraphs. Currently no consensus exist in the field of obstacle avoidance on how to measure the performance of a system. This is remarkable, considering the countless research contributions done in the field. Obstacle avoidance is one of the fundamental problems in robotics and has been a research field since the robot was invented. So what makes it so difficult to come up with such an evaluation method?

A reason why it is difficult to develop a standardized evaluation method is that performance of an OA algorithm depends on the environment in which it operates. This environment can be complex and difficult to describe. This challenge is also stated in a survey from Kuchar & Yang (2000):

Additionally, a consistent benchmarking method for analyzing and validating models is required. This is difficult due to the variety of operational modes and conditions to which Collision Detection & Resolution (CDR) systems may be exposed, but will be necessary in order to select the most effective systems for implementation in the field.

The variety of operational modes is also seen in the field of face recognition. In which the problem is known as the Pose, Illumination and Expression variant problem. Even for three characteristics it is difficult to determine the performance. For OA one could easily think of more than three characteristics. This combined with the fact that researchers are not always interested in the complete performance, probably explains why no consensus currently exist in evaluating OA methods.

In computer vision the multidimensionality problem is often solved by using a set of images which capture a diversity of conditions. An example of such a data-set is the Middleburry set. Unfortunately such data-sets are not very useful for OA. Since no direct relation is made between the environment condition and the performance. Due to the complexity of the environment it is not feasible to create a complete data set with all conditions and therefore a method is needed which can predict the performance for untested environments. To do this a test is needed in which structured environment changes are made, but due to the variety of conditions it might be hard to describe these changes.

Another factor which makes it difficult to develop a standard evaluation method is that different OA methods are sensitive for different environments. A system which detects obstacles with a radar is able to operate in different conditions than a system which uses stereo-vision to detect obstacles. So different OA methods need different evaluation methods.

A third challenge can be seen in figure 2-1. In the figure the main control loop of an obstacle avoidance system for an Unmanned Aircraft Vehicle System (UAV) is shown. The overall performance of the system is the combination of all these blocks and how these interact with the environment. It is difficult to decouple the blocks, especially in real flight. For example, when researcher use different UAVs to demonstrate their OA algorithm, it becomes difficult to specify if the performance difference is caused by the OA algorithm or caused by the dynamical constraints of the UAV.

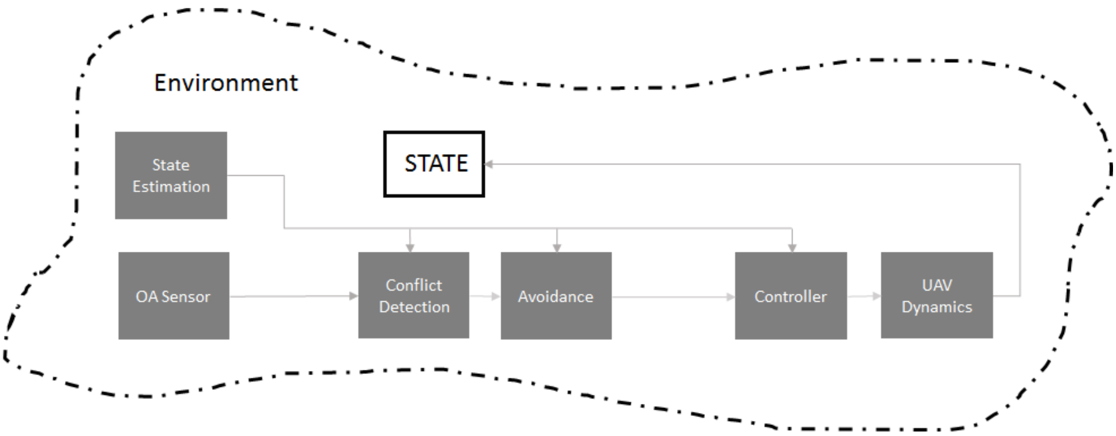


Figure 2-1: Obstacle avoidance control loop

To summarize, three challenges exist in the development of a standardized evaluation method:

1. Variety of operational modes and conditions.
2. Different OA methods require different evaluation measurements.
3. Total performance depends on the complete control loop.

To develop an evaluation method two thinks needs to be defined: environment metrics and performance metrics. Environment metrics define the environment in which an experiment is done while the performance metric specifies what is measured in this environment. Specifying the environment metric is directly related to the challenges described previously.

Also practical issues need to be considered when developing a new evaluation method. Testing takes time, which most researchers do not have. To come up with a method that is widely accepted the method should be time efficient and the cost should be manageable.

To find the needed environment and performance metrics a broad overview of current OA systems needs to be created, such that all options are taken into account. From this search experiments can be designed and executed.

2-2 Research Questions

From the previous sections the need and challenges for an evaluation method become clear. The goal of this research is to develop such a method which can be stated as follows:

The objective of this research is to develop a general evaluation method for obstacle avoidance. The proposed method will be used to select an obstacle avoidance method for a drone serving application.

So the objective has two parts, one part is to define a method to measure the performance and another part that selects the best system for the restaurant application. The performance part is split into two parts as well. A detection part and an avoidance part. This division is based on the work of Albaker & Rahim (2010) and will be discussed in detail in chapter 3. The three central questions become:

1. What is the performance of current detection systems?
2. What is the performance of current avoidance systems?
3. What is the best system for the restaurant application?

These three central questions can be divided into sub-questions. Question one and two have similar sub-questions, since to answer both questions a similar approach is needed. Both have three sub-questions in which first an overview of current system is needed after which the relevant metrics need to be determined. The final step is to find the relation between the algorithm and the metric of interest.

1.1 Which detection algorithms are currently available?
1.2 Which performance and environment metrics are relevant for assessing detection performance?
1.3 What is the relation between the environment and performance metrics for the available algorithms?

2.1 Which avoidance systems are currently available?
2.2 Which performance and environment metrics are relevant for assessing avoidance performance?
2.3 What is the relation between the environment and performance metrics for the available algorithms?

The third central question is answered by first answering the question what kind of environment the restaurant is. After which the question is answered which algorithm is the best fit for this environment.

3.1 In what type of environment is the restaurant operating?
3.2 Which system performs best in this environment?

The following section will focus on sub-questions 1.1, 1.2, 2.1 and 2.2 in which an overview is given of the available literature in this field.

Chapter 3

Literature overview and general evaluation considerations

In this section an overview of the relevant literature is given to find answers to the first sub-questions stated in section 2-2. First the terminology is discussed after which an overview of the field of obstacle avoidance is given. The chapter is finalized by discussing what aspects influence the OA performance.

3-1 Terminology

Before discussing the literature some definitions need to be cleared out. In the literature the same words are used with different meanings, which can cause confusion. The following explains how these terms are used in this survey.

3-1-1 General terminology

Obstacle Avoidance, Collision avoidance In this survey the term Obstacle Avoidance is given to the complete research field in which autonomous vehicles avoid obstacles of any kind. Collision avoidance is a subset of OA in which the obstacles only consist of other vehicles. Later in this section collision avoidance will also be referred to as obstacle avoidance with intelligent agents.

Detection, Conflict detection Detection and conflict detection are sometimes interchanged in literature. In this survey detection is the process of measuring the state of an object. Conflict detection is done in some OA algorithms in which the state of the agent is predicted and compared to the predicted state of the obstacle. Detection is the input for conflict detection.

Avoidance, Resolution As will be seen later in this survey, OA consist of two main parts, detecting obstacles and performing an avoidance maneuver. This later term is sometimes referred to as the resolution method. Here 'avoidance method' will be used instead of resolution, since it seems more fit. This can be confusing when it is used in combination with the overarching term OA. But for the reader it should be clear which of the two is meant by its context.

Guidance, Navigation Guidance and navigation are often intertwined, it can therefore be confusing when the terms are used independently. In this survey navigation is the process of determining the state of an UAV and the state(position, direction) of the goal. Guidance is the process to determine the path to get from the current state to the goal.

3-1-2 Avoidance Terminology

Reactive, Non-reactive Reactive control is a system in which the sensor inputs and control outputs are tightly coupled. No extensive calculations are performed. Reactive control generally has no memory or internal representation of the world. In Obstacle Avoidance (OA) this term is used in two ways. In some cases the term is used to state that only the current measurements of the sensors are used to come up with a control output. In the second option the term is used to state that no internal model of obstacles is present, instead the inputs are directly coupled to come up with a control output. In this survey the second/broader definition of reactive systems will be used.

Online, Offline Online means a calculation is done in real-time, while offline algorithms are run before a flight takes place. The terms are used to distinguish online mapping (such as Simultaneous Localization and Mapping (SLAM)) from offline mapping and online path planning from online path planning.

Local planning , Global planning Local planners are online planners in which the planned path is constrained to a certain distance or time. A global planner computes the whole path from the initial state to the goal state. From this it can be concluded that a global planner is always a system in which OA and guidance are combined.

3-1-3 Relation between terms

Another cause for confusion is that terms are related. But sometimes relations are drawn which are not valid. These relations can be seen in Figure 3-1. The full lines represent valid relations while the dotted lines represent invalid relations.

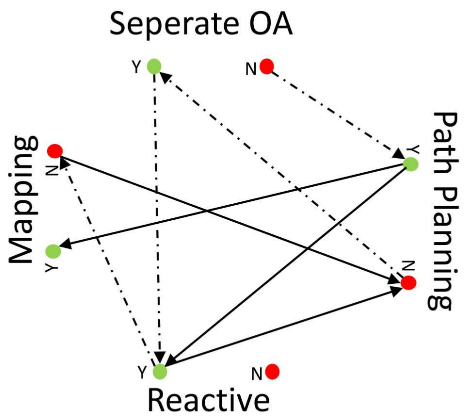


Figure 3-1: Relations between obstacle avoidance terminology

A first distinction can be made between systems in which guidance and OA are separated and systems in which they are combined. The last case is sometimes referred to as online path planning, as can be seen by the dotted arrow, but in this survey a distinction will be made between a combined system and online path planning. The reason for separating the two terms is because one could easily think of a system which plans a path around an obstacle in real time while still having a separate guidance system.

The second term in the figure is path planning. An OA system that uses a path planner needs a map. So by definition a path planner also has a map which can be developed online or offline. The converse is not true, if a system has a map it does not necessarily use a path planner.

The third term is mapping in this survey mapping is used in the context of obstacle avoidance. So mapless algorithms are methods which do not use a map for obstacle avoidance. If such a system has a separate guidance algorithm a map could still be used to create an initial path.

The final distinction is made between reactive and non-reactive methods. A reactive method is by definition not a path planner. Often a reactive method is used in combination with an offline global path planner, but this does not have to be the case. Another invalid relation which is drawn is that a system with a separate OA algorithm is always reactive. Readers must be careful in making assumptions about a system when seeing these terms.

3-2 Techniques for algorithm performance analysis

In section 2 it was stated that environment metrics and performance metrics need to be defined to create a standardized evaluation method. These can be used for different types of evaluation techniques.

A broad distinction can be made between empirical and theoretical evaluation techniques. In the first case the goal is to find a theoretical limit in the second case an experiment is done to find the performance.

Another difference can be made between technology evaluation and scenario evaluation. In technology evaluation one tries to find the fundamental limits of a system. In scenario evaluation the researcher only looks at specific cases in which the system will operate. In this survey the choice is made to create an empirical technology based evaluation method.

Another choice needs to be made between real flight-tests and simulations. To understand the difference between the two tests one could again look at figure 2-1. In the figure the main loop of an OA system is shown. In a real flight-test the response is the result of the complete loop. So the performance depends on all blocks in the loop. For simulations the influence of detection can be mitigated by assuming perfect sensing. Also perfect state estimation can be assumed making the system more robust. These assumptions cannot be made for real flight-tests in which sensor noise and disturbances are always present. This noise might be the biggest challenge of the OA system. Therefore assuming perfect sensing or perfect state estimation might result in performances which are not comparable to real life scenarios. To understand what the biggest effect on the performance of an OA system is, both simulations and real flight-test should be done.

3-3 Overview of obstacle avoidance methods

To come up with a standardized evaluation method an overview of the state-of-art in obstacle avoidance is required. What are the methods currently applied and how are those methods being evaluated? It is difficult to give a complete literature survey because of the vast amount of work already done in obstacle avoidance. Or as stated in (Bonin-Font et al., 2008):

The scope of robotics as a discipline and the huge number of existing contributions make it almost impossible to make a complete account

Nevertheless an overview can be created. A good starting point is an overview created by Albaker & Rahim (2010) which breaks an OA system down into five 'design factors' similar to the ones shown in figure 2-1 namely; sensors, conflict detection, resolution, maneuver realization and general design factors. A schematic representation is given in 'Figure 3-2.

In the figure the first factor is the sensor. The selection of the sensor has an influence on what data is available. The second branch uses this raw data to detect conflicts. Part of this branch is state propagation which is common in air traffic management. The next step of the obstacle avoidance system is to find an escape trajectory such that the

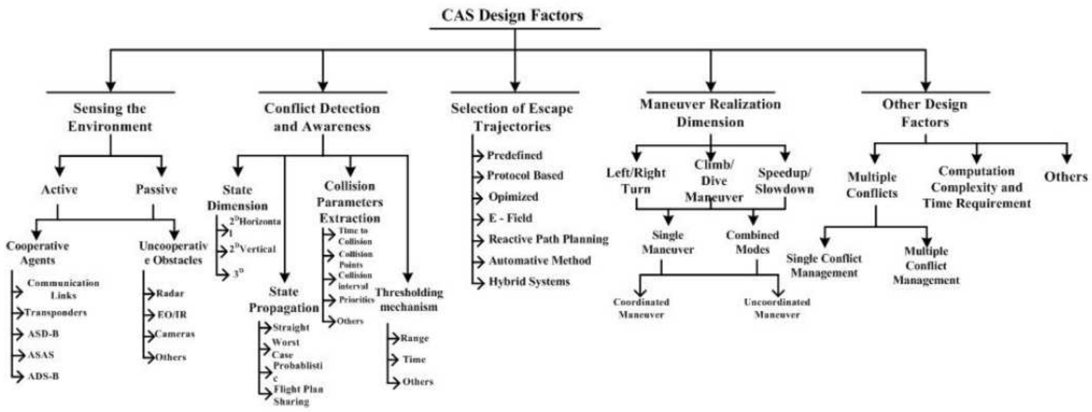


Figure 3-2: CAS Design Factors

Micro Air Vehicle (MAV) avoids the detected obstacles. The final step is to realize this path with a certain control sequence. The scheme is finalized with factors that have an impact on the OA system, but do not have a direct function.

As said the paper gives a nice overview but is written from a collision avoidance perspective. In this study a general perspective for obstacle avoidance is taken. The main difference is that in collision avoidance the obstacles are always dynamic and the time between detection and resolution is generally larger. For example for Traffic Alert and Collision Avoidance System (TCAS) systems the time between detection and resolution can be several minutes while for a UAV flying indoor the time to avoid an obstacle is in the order of seconds. Another issue that is not discussed by Albaker, is the type of environment. How many obstacles are there for example, in a dense environment one might want to use a different obstacle avoidance strategy than when the obstacles are sparse.

Another overview is created in the work of Kuchar & Yang (2000). Kuchar makes a distinction based on five properties; Dimensions, Detection, Resolution, Maneuvers and Multiple. Similar to the division set out in Figure 3-2. For several algorithms the type in each property is stated, which can be seen in Appendix V. Kuchar also looks to obstacle avoidance from an Air Traffic Management (ATM) context and does not consider all properties of an OA system (the sensors for example). In this survey an attempt is made to give a complete overview in which all aspects of an OA system are considered.

Subdividing OA algorithms can be done in many ways Desouza & Kak (2002) divides it between indoor and outdoor approaches. Another division is made by Goerzen et al. (2010). Goerzen makes a distinction between algorithms which can cope with constraints or algorithms which cannot. Goerzen compares the algorithms in each category by listing them in a table. The characteristics in the table are: number of dimension, completeness, optimality and proven time complexity. The completeness of an algorithm means if a solution is always found(if it exist). Optimality specifies if the found solution is optimal,

Table 3-1: OA characterization

	Basic			*Environment*					*Sensors*			*Conflict detection*						*Avoidance*			
	Single(1)/Multiple(2)	Combined(1) / Seperate OA(2)	(online) Mapping(1)/ No mapping(2)	Unknown(1)/ Known(2)	Static Obst. Yes(1)/No(2)	Dynamic Obst. Yes(1)/No(2)	Intelligent Obst. No communication Yes(1)/ No(2)	Intelligent Obst. With communication Yes(1)/ No(2)	Sensor type(s)	Method to calculate states	States measured	Explicit conflict detection Yes(1)/ No(2)	Dynamic constraints Yes(1)/ No(2)	Obstacle uncertainty Yes(1)/ No(2)	Collision flag(1)/ Collision set(2)	State dimension	State propagation	Dynamic constraints Yes(1)/ No(2)	Multiple(1) /Single(2) obst. Taken into account	Avoidance method	Control actions(velocity/turn/both)
Fiorini1998	1	1	2	1	1	1	2	2			Pos, vel	1	2	2	2	2D	Worst Case	1	1	Heuristic Search	Velocity Vector

semi-optimal or non-optimal. Finally the order of complexity is expressed using the big O notation. Bonin-Font et al. (2008) uses yet another division. Bonin-font divides it between mapping and mapless algorithms. Bonin-Font et al. (2008) also make a distinction between the type of vehicle and the type of vision strategy that is chosen. The tables constructed by Goerzen et al. and Bonin-Font et al. are given in Appendix V and Appendix V.

Similar to the table of Bonin-Font et al. a table is constructed in this survey to get insight in the available methods for obstacle avoidance. The way it is divided in this survey can be seen in Table 3-1. Table 3-1 was constructed after reading 143 papers on obstacle avoidance. The first part of the table states the basic 'top level' distinction between OA algorithms and the type of objects for which the algorithm was constructed. The first division is made between single agents and multiple agents. Another distinction is made between systems in which the guidance and obstacle avoidance are combined and systems in which the guidance and obstacle avoidance are separated, which was also discussed in section 3-1. The distinction between these two categories was also made by Mujumdar & Padhi (2011). Who stated it as follows:

The problem of avoiding collision with obstacles online can be perceived in two ways: as a path-planning problem, and a collision-avoidance maneuvering problem.

Another 'basic' distinction is made between systems which use data from their sensors to build an (online) map and systems in which no real-time map is being build. The later case does not necessarily mean no map is present. A system could still have an (offline) map of permanent obstacles but does not update it in real-time.

Next to the basic or top-level distinctions between obstacle avoidance systems there is also a difference in the type of environments for which the systems are designed. First there are systems in which the environment is known. In which the problem often boils down to a path planning problem. For unknown obstacles there are two types of objects 'static', and 'dynamic'. Dynamic obstacles can be subdivided into Intelligent and non-intelligent obstacles. With intelligent obstacles a system is meant that also has an obstacle avoidance system, which can interact with the OA system of the UAV. The way in which these OA systems interact can cause problems and therefore these objects are not the same as 'normal' dynamic objects. It is assumed that static obstacles are always non-intelligent. The dynamic intelligent obstacles are divided into obstacles with communication' and 'obstacles without communications'. Intelligent obstacles with communication are typically seen in ATM on which the papers form Albaker and Kuchar were focused.

Table 3-2: Characterization of performance evaluation

	Detection performance			Avoidance Performance									
	Detection performance	Environment type	Performance metric	Real(1)/ Simulated(2)/ Both(B)	Indoors(1)/ Outdoors(2)	Dense(1)/ Sparse(2)	Static Obst. Yes(1)/ No(2)	Dynamic Obst. Yes(1)/ No(2)	Intelligent Obst./ No communication Yes(1)/ No(2)	Intelligent obst. With communication Yes(1)/ No(2)	# tests	Testing	Performance parameter
Fiorini1998						1	1	2	2	static+dynamic obstacles			Successful avoidance

Other parts of the table are 'sensors', 'conflict detection' and 'avoidance'. These parts will be discussed in detail in chapters 4 and 5. Besides the characterization of the method, also the performance analysis of each paper is characterized. This can be seen in Table 3-2. Also this part will be discussed in more depth in chapters 4 and 5. The full table is given in Appendix V.

3-4 Performance incentives

Now a general overview of the field of obstacle avoidance is present one could focus on what influences the performance of an OA system. To answer this question one could again look at the general OA loop from figure 2-1. The combination of these blocks determine the overall performance of the system. To analyze the overall performance one could try to split the performance measurement into individual blocks. This can be seen in figure 3-3.

In the figure four blocks are identified for which the performance could be determined independently. Block A represents the 'detection' part of an OA algorithm, Block B represents the determination of the avoidance maneuver. The avoidance maneuver is determined by the combination of conflict detection and avoidance. This conclusion was made using the overview discussed in the previous section. Many algorithms do not

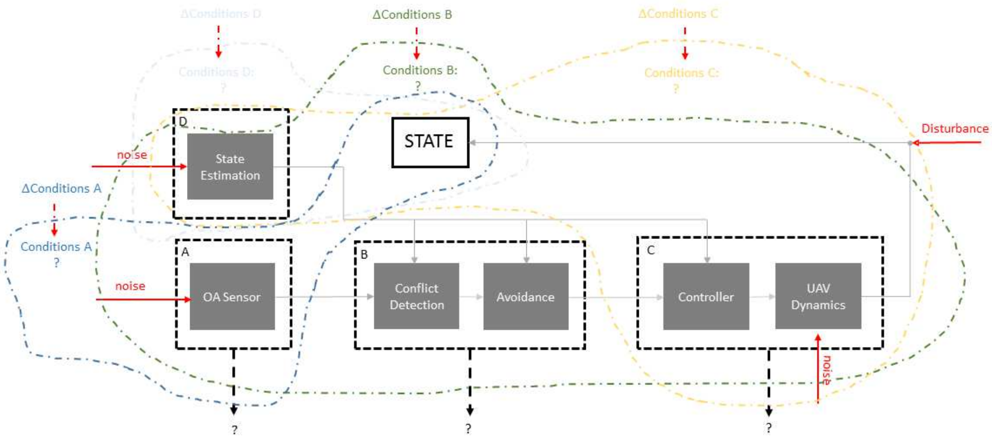


Figure 3-3: Influence on OA performance within the OA loop

Table 3-3: Performance dependencies in OA loop

Block:	Dependent on:
A	State
B	A,C,D,State
C	D, State
D	State

have an explicit conflict detection method, but if a system does have explicit conflict detection it is often strongly connected to the way in which an avoidance maneuver is found. Block C consist of the controller and the UAV dynamics. Finally Block D consist of the state estimation.

For each block the performance can be analyzed. For example for Block C, the performance could be analyzed using classical control theory. Although the performance of Block C could be analyzed independently from other blocks, it would require input from a state estimator. Therefore some assumption needs to be made on the performance of the state estimator. These dependencies are visualized in figure 3-3 by including the dependent blocks in the 'environment'(dotted line) of the block for which the performance needs to be determined. For example the state estimation is included in 'environment C' and therefore some assumptions need to be made about state estimation when determining the performance of Block C. The dependencies are also listed in table 3-3.

This thesis will focus on determining the performance of Block A and Block B, since Block C and D are not directly a part of an OA system. In essence the performance of an

OA system comes down to the performance of the detection part and the performance of the avoidance part. Such a distinction was also made by Y. Kwag & Kang (2004) which stated the problem as follows:

One is the awareness problem, the other is the avoidance problem.

For the detection part(Block A) it can be seen in table 3-3 that no assumption need to be made about the other blocks, but only about the state. It is for example more difficult to detect obstacles at a larger distance or at a larger speed. For Block B assumptions need to be made for Block A, C and D, since the performance of Block B is dependent on all other blocks.

Besides the dependencies on other blocks the performance is also dependent on the environment. The environment which influences the performance of each block is different, therefore a different set of environment conditions is shown for each block in figure 3-3 . For example the environment of Block A is represented by the blue line. The environment for which it is sensitive are the conditions A in figure 3-3. The final influence on the performance is the amount of noise present in the loop. This noise can enter the loop directly through sensor noise or disturbances. But also indirectly through the environment which is represented by the dotted red arrows.

Unfortunately the environments for which Block A and B are sensitive are unknown, which is represented by the colored question marks. Also the metric to measure the performance is unknown which is represented by the question mark under the black dotted line.

To come up with a generalized method an answer needs to be find to these question marks. The question mark are directly related to the sub-question sated in section 2-2. Sub-question 1.2 is equivalent to filling the question mark below block A and finding conditions A. While sub-question 2.2 is equivalent to filling the question mark below block B and finding conditions B. In the following chapters these parts are discussed in detail. First the detection problem will be addressed, after which conflict detection and avoidance methods are discussed.

As a final note, the answer to the previous questions depend on the type of obstacle avoidance method which is used. Which was stated in section 2-1 as: different OA methods require different evaluation measurements. Therefore an overview of the available methods is needed for which the table discussed in the previous section can be used.

The methods chosen in the detection part and the avoidance part determine how the previous questions are answered. Not only the specific method used in detection or avoidance determine the environment and performance metrics but also the 'top-level' choices which were shown in Table 3-1. An OA system which can cope with dynamic obstacles for examples needs a different set of environment metrics then a system which can only cope with static obstacles.

Chapter 4

Detection

The fist step in obstacle avoidance is detection. The performance of an obstacle avoidance system depends on how well it can detect obstacles. In this section the methods for detection are set out and their performance limitations are discussed.

To have any detection at all a system needs some sort of sensor. To get insight into different types of sensors, one can again look at Figure3-2. In which Albaker & Rahim (2010) makes a distinction between active an passive sensors. Albaker also makes a distinction between techniques developed for cooperative agents and systems for uncooperative obstacles. Examples of detection techniques for cooperative agents are ASAS or ADS-B. This literature review focuses on detection methods for uncooperative obstacles. Since the UAV serving application needs to be able to maneuver around uncooperative objects. Uncooperative sensors can be divided into five main methods: vision, Infrared Sensor (IR), ultrasonic, laser and radar.

The choice for which sensor to use depends on the application. Different types of sensors have different characteristics. For example the weight, cost, power or range vary between sensors. Each sensor has different limits and is sensitive to different environments. In the following the physical principle, the performance limits and OA implementations for each sensor are discussed.

4-1 Vision

One major field in detection is vision. Many books have been written about computer vision (Young et al., 1998), (Davies, 2012), (Gool et al., 2011). To determine the performance of vision based algorithms a research was conducted by by Thacker et al. (2008). In his survey Thacker gives an overview of the performance characterization methods used in computer vision. The paper lists the best practices of performance characterization of several fields in computer vision. Thacker evaluates the performance characterization by answering five closed questions which are given in Table 4-1.

Table 4-1: Summary Thacker et al.

	Is there a data set for which the correct answers are known?	Are there data sets in common use?	Are there experiments which show that the algorithm works as expected?	Are there any strawman algorithms?	Is there a quantitative methodology for the design of algorithms?
Sensor characterization	Yes	Yes	Yes	N/A	Yes
Video compression	Yes	Yes	Yes	Yes	No
Feature detection	(Yes)	Yes	(Yes)	Yes	(Yes)
Object localization	Yes	No	(Yes)	No	(Yes)
Object indexing	Yes	(Yes)	(Yes)	No	(Yes)
Optical flow	Yes	(Yes)	Yes	Yes	(Yes)
Stereo vision	Yes	(Yes)	(Yes)	Yes	(Yes)
Face recognition	Yes	(Yes)	(Yes)	(Yes)	No
Medical structure	Yes	Yes	Yes	(Yes)	(Yes)

The first step suggested by Thacker to characterize vision sensors is to consider the illumination regime in which it operates. But other aspect of the system might influence the performance of the sensor as well, such as: noise, lens distortion, sensor deformation. Not all fields of computer vision listed in table 4-1 are useful for obstacle avoidance. Only optical flow and stereo vision algorithms are currently being used in OA systems. Generally optical flow is used when one camera is used and stereo vision is used when two camera's are present. In the following these two options are discussed in detail.

4-1-1 Monocular vision

principle In monocular vision a single camera is used to detect obstacles, using some form of Image Processing (IP). The advantage of this system for UAV applications is that currently a lot of UAVs are equipped with a camera and no hardware changes need to be made. To detect objects from a single camera several strategies can be chosen. A nice overview is given in the work of Guzel & Bicker (2009) who defines two fundamental groups of vision based obstacle avoidance. Namely optical flow based techniques and appearance based methods.

Appearance based methods are based on the appearance of individual pixels. Only the information available within one image frame is used. One appearance based method to detect obstacles for ground robots is to detect pixels which differ from the ground and classify them as objects. To use this method the ground needs to be homogeneous and relative flat. A disadvantage of this method is that difference in the ground color are detected as obstacles as well. For UAVs the sky can be used as the background as was done by Croon, G.C.H.E. de (2011).

Another monocular vision technique is optical flow. Optical flow based techniques are based on the assumption that the intensity of objects in a scene remain constant. Which is known as the 'conservation of image intensity'. When an object is moving(or the camera is moving) the assumption can be used to calculate the velocity of an object.

This can be done using equation 4-1.

$$\frac{\delta I}{\delta x}\Delta x + \frac{\delta I}{\delta y}\Delta y + \frac{\delta I}{\delta t}\Delta t = 0 \tag{4-1}$$

In this equation I represent the Intensity of an object, δx and δy represent the position difference in the image frame and δt represent the time difference. The time difference δt is known based on the frame rate, leaving two unknown variables: δx and δy. Which is known as the aperture problem. Only one equation is available so additional assumptions need to be made to solve the problem. Many methods exist to solve this problem, generally these can be subdivided into four techniques: Differential Techniques, Region-based matching, Energy-based methods and Phase-Based Techniques. A detailed description of these methods can be found in Barron et al. (1994). Two widely used methods are Lucas-Kanade and Horn-Schunck, both are differential methods. Lucas-Kanade solves the aperture problem by assuming the flow in a small pixel window is constant. Horn-Schunck solves the problem by assuming a certain amount of smoothness over the complete image. Optical flow methods can be easy to implement but remain sensitive to illumination conditions.

Performance To analyze the performance of the above described methods a look can be given to the paper from Thacker, which evaluates performance metrics for optical flow. According to Thacker current methods evaluate optical flow by measuring the mean error in magnitude or direction. Thacker advised to also include covariance matrices to quantify the performance.

Implementation Many researches have proven to be able to successfully avoid obstacles using monocular vision. When looking at the table in Appendix V several monocular techniques can be found. In the table no distinction between the monocular methods is being made. Examples of appearance based methods are Magree et al. (2014), Saha et al. (2014) and Huang et al. (2006) who uses feature extraction, to determine the heading and width of obstacles. Unfortunately none of these papers evaluate how well the algorithm detects the obstacle.

The same holds for optical flow, several implementations are present but none have a separate detection evaluation. Marlow & Langelaan (2011) use optic flow to determine range estimates to obstacles. Marlow compares the estimates with the 'ground truth' but does not state any quantitative value. Gosiewski et al. (2011) implements an optical flow algorithm by making use of the Lucas-Kanade algorithm. In another paper from Sebesta & Baillieul (2012) optic flow is used to estimate the time to collision. The implementation from Stowers mentioned in the introduction, also makes uses of optical flow, but does not measure the performance of the algorithm either.

4-1-2 Stereo Vision

Principle Instead of using one camera also two cameras could be used to detect obstacles. The major advantage of such systems is that it is possible to directly measure the distance to an object. The distance can be calculated by using Equation 4-2.

$$\Delta d = \frac{f \cdot b}{Z} \tag{4-2}$$

In Equation 4-2 Δd represents the pixel displacement, f is the focal length of the camera (in pixels), b is the distance between the two camera's(baseline) and Z is the unknown distance to the obstacle.

To get an overview of the available algorithms in stereo vision a look is given to a survey paper from Lazaros et al. (2008). Lazaros points out the vast amount of research effort currently employed in stereo vision which makes it difficult to keep up. Since a complete survey is difficult and not completely relevant in this discussion, only the main methods of stereo vision will be discussed.

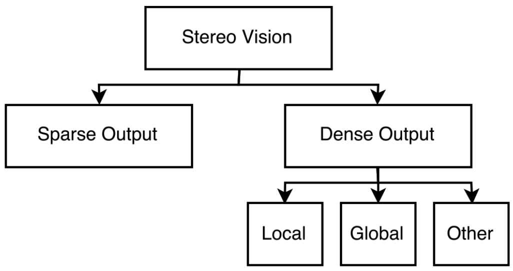


Figure 4-1: Categorization of stereo vision algorithms

Lazaros evaluates software and hardware implementations, here only the software implementations will be discussed which can be seen in Figure 4-1. The software implementation is split into sparse outputs and dense outputs.

Algorithms resulting in sparse outputs focus on features in the images, such as corners or lines. Poorly textured areas remain unmatched. Because not every pixel is being matched higher processing speeds are possible.

Dense algorithms can be divided into local and global methods. Local methods calculate the disparity of each pixel according to the information provided by its local, neighboring pixels. Global methods use the whole image to calculate the disparity. Lazaros splits the local methods on the basis of which cost function is used to solve the correspondence problem. The correspondence problem is the problem to find for each point in the left image the corresponding pixel in the right image, which is often solved using a Sum of Absolute Differences (SAD) cost function.

Global methods produce more accurate results than local methods. The goal is to find the minimum of a cost function (similar to local methods), but now the whole image is taken into account. Because an optimization algorithm needs to be run for every pixel global methods are computationally more demanding. Also for global methods a subdivision is made. Namely into those performing a global energy minimization and those pursuing the minimum for independent lines. For both local and global methods the categorization is completed by specifying if the algorithm makes use of color instead of a gray-scale and by specifying if the method uses some form of occlusion handling.

Performance Stereo vision has some fundamental performance limits, which are stated in a paper from Kytö et al. (2011). The theoretical stereo resolution is given in Equation 4-3

$$dZ_c = \frac{Z^2}{f \cdot b} dp_x \tag{4-3}$$

In equation 4-3 the variables Z, f and b are the same as in Equation 4-2, dp_x represents the minimum disparity difference that can be measured. Generally this disparity resolution is one pixel but also methods exist which make use of sub-pixels in which the minimum disparity is smaller than one pixel. From this formula it can be concluded that the resolution decreases with increasing distance which is visualized in Figure 4-2. The larger the distance the bigger the diamond shape, meaning a lower resolution.

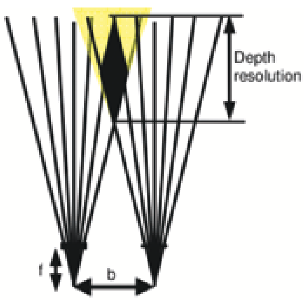


Figure 4-2: Visualization of depth resolution

When the baseline or the focal length becomes smaller the resolution degrades. These limits should be taken into account when designing a stereo vision system. The limit represented in Equation 4-3 is the theoretical limit, other factors such as misalignment and calibration will make the resolution even lower. For an elaborate evaluation between the theoretical limits and the practical limits the reader is referred to the work of Kytö et al. (2011).

Another theoretical limit is the maximum range as can be seen in Figure 4-3. In this figure the disparity is converted to a distance using Equation 4-2. A baseline of 60 mm and a focal length of 181 px are used. As can be seen the distance goes to infinity for a

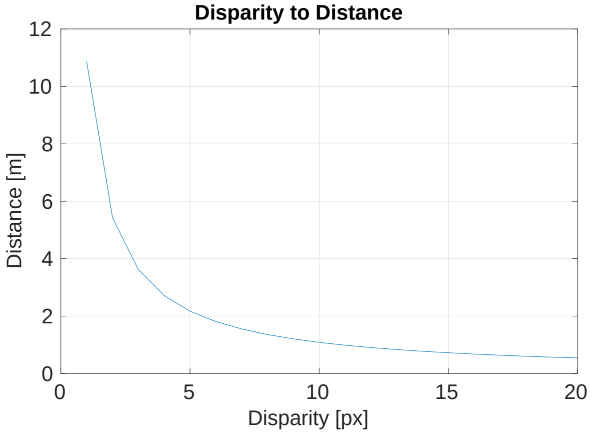


Figure 4-3: Distance as a function of disparity

disparity of zero pixels. The maximum range can be calculated by putting the minimum disparity in Equation 4-2. In this situation a minimum disparity of one pixel is assumed giving a maximum range of $11m$. In the figure also the decreasing resolution can be observed by looking at the gradient. The gradient increases for large distances meaning a lower resolution.

Also in literature the performance of stereo vision is discussed for example by Sabater et al. (2011), which evaluates the disparity error for several stereo algorithms. It compares stereo methods by calculating the Root-Mean Squared (RMS) error. The results are compared to the predicted error based on the estimated noise present in the image. As shown by Sabater, the type of algorithm can have an influence on the RMS error and thus on the OA performance of the system.

A paper by Lazaros et al. (2008) also reflects on the performance of stereo vision algorithms. The most common used image set to evaluate stereo-vision is the Middlebury data-set. The performance metric on this data-set can differ, but according to Lazaros the preferred metric of evaluation for stereo vision is the percentage of pixels whose absolute disparity is greater than one, in the unoccluded areas of the image. Lazarus gives a nice overview of the values for several algorithms executed on the Middlebury data-set, no direct comparison is being made but the disparity error ranges form 0.9% to 7% for local methods and from 0.1% to 7% for global methods. To complete the performance evaluation of the algorithms, Lazarus states the frame rate, with the processor on which the calculation was performed.

Another research which makes use of the Middlebury data-set is conducted by Scharstein & Szeliski (2002). Scharstein gives an overview of the available algorithms and evaluates them thoroughly. The metric used to evaluate the performance is the Root-Mean Squared (RMS) error, for which the disparity map is compared to the ground truth. Another metric used is the percentage of bad matching pixels, which was the preferred metric for Lazaros et al. (2008).

To evaluate the performance, Scharstein calculates these metrics for different type of regions. Three regions are distinguished, textureless regions($\mathcal{T}$), occluded regions($\mathcal{O}$) and depth discontinuity regions($\mathcal{D}$). Also the complements of these regions can be used.($\bar{\mathcal{T}}, \bar{\mathcal{O}}, \bar{\mathcal{D}}$) The performance of 20 algorithms using these metrics and regions is shown in Appendix V.

Also another approach of determining the performance of stereo-vision algorithms is mentioned by Scharstein which is know as the prediction error strategy. The method evaluates stereo algorithms by predicting a third image using the disparity map. This predicted image is than compared to a third image taken with a camera at a different baseline. The error between the two images serves as a metric to define the performance of the algorithm. The advantage of this method is that no ground-truth is needed, but only an extra set of images. The method is applied by Mahmood et al. (2012). In which stereo-vision algorithms are analyzed under three different light conditions.

Finally one could look at Table 4-1, in which the survey from Thacker et al. (2008) is summarized. In the survey the Middlebury data-set is mentioned again as a common used data-set. But Thacker also mentioned that other data-sets are available and a consensus still appears to be missing on which set to use. Another relevant finding in stereo vision performance by Thacker is stated as follows:

No work appears to have been carried out on relating the characteristics of a data set which would allow the performance on unseen data to be estimated.

Meaning the performance measurements applied on the Middlebury data-set cannot be used to estimate the performance on other data-sets. In this thesis the goal is to develop a method which would allow to make such estimations.

Implementation Five implementations make use of stereo vision in Appendix V. One method takes the performance limits of the stereo detection into account (Frew & Sengupta, 2004). In this method an unsafe region is established, part of this unsafe region is a diamond shaped region resulting form the error of the stereo system. Other methods use the stereo vision to build a map (Hrabar, 2008), (Hrabar, 2011),(Jongho & Youdan, 2012) but do not consider the detection performance.

4-2 Infrared

Principle In the previous section visible light is used to detect obstacles, also other domains of the electromagnetic spectrum can be used such as infrared. To detect obstacles using infrared both active and passive options are possible. In the passive case an infrared camera is used to collect images in the infrared spectrum. Similar techniques as the ones described in the previous section can be applied to these images. Also for infrared cameras monocular and stereo vision is possible.

Another simpler form of Passive Infrared Sensor (PIR) exist. Such sensors only consist of a photo-diode which converts the energy in the infrared domain to a voltage. Such sensors are widely used in alarm systems or light switches.

An active infrared sensor also has a photo-diode to detect infrared radiation, but works on a different principle. Besides the photo-diode a LED is present emitting at the infrared frequency. The waves sent from the LED will reflect back from an object resulting in a peak in the detector. From the time difference between the emitted pulse and the measured peak the distance can be calculated.

$$y = \frac{\alpha}{x^2} cos\theta + \beta \tag{4-4}$$

To determine the distance between the IR sensor and an object Equation 4-4 can be used, y is the sensor output, x is the distance, α is a constant which includes the radiant intensity, spectral sensitivity, the gain of the amplifier and the reflectivity of the target. The last term θ is the angle of incidence of the obstacle.

Performance As concluded in the work from Benet et al. (2002), uncertainties in any of the values in Equation 4-4 result in a distance error. It is assumed that α an β can be determined with sufficient precision. The two remaining errors sources are the noise in the measurement ϵ_y and the uncertainty in the angle of incidence ϵ_θ. Both can be related to the total error using Equation 4-4, which is given in Equations 4-5 and 4-6. In this equation ϵ_x represents the distance error, ϵ_{xy} represents the distance error due to measurement noise and and $\epsilon_{x\theta}$ represents the distance error due to the uncertainty in the angle of incidence.

$$\epsilon_x = \epsilon_{xy} + \epsilon_{x\theta} = \frac{\delta x}{\delta y}\epsilon_y + \frac{\delta x}{\delta \theta}\epsilon_\theta \tag{4-5}$$

$$\epsilon_x = -\frac{x^3}{2\alpha cos\theta}\epsilon_y - \frac{x}{2} tan\theta \cdot \epsilon_\theta \tag{4-6}$$

It can be seen in Equation 4-6 that the error caused by measurement noise grows cubic with respect to the distance. To mitigate the error of the unknown angle Benet provides a method to estimate θ using measurements of multiple sensors. Since there is noise in the sensors an uncertainty will be present in the estimation of θ, which results in a distance error. The effect of the noise is dependent on the distance and inclination of the obstacle, which can be seen in Figure 4-4.

The figure represents the relation between the incidence angle and the mean distance with respect to the RMS error. A noise level of ($\sigma_y = 6mV$) is assumed. The figure shows that the error increases rapidly with respect to the distance. At a distance of only $100cm$ the error already becomes around $10cm$, which is large.

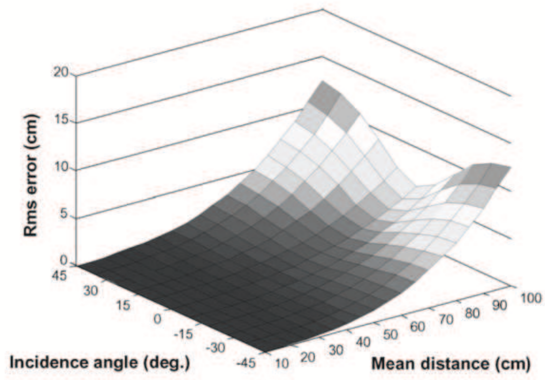


Figure 4-4: Measurement error infra-red sensor

Implementation Because of this increasing error, active infrared sensors, are often only used for short ranges or as a complementary sensor. An example of such a system is presented in (Fasano et al., 2006).

4-3 Ultrasonic Range Finders

Principle Infrared detection systems can be passive or active, purely active systems are ultrasonic sensors. Ultrasonic range finders work similar as active infrared. A (ultra)sound is transmitted after which the receiver 'listens' for the reflected sound signals. The strength of the received signal is mapped to a distance between the sensor and the obstacle. Ultrasonic sensors are light and do not require a lot of computational power.

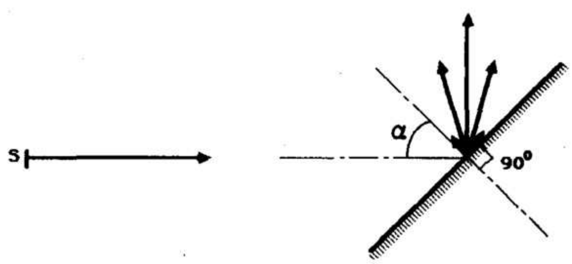


Figure 4-5: Reflected sound waves on an inclined obstacle

Performance A disadvantage of ultrasonic sensors is interference. Multi-paths but also other UAVs can cause interference in the signal resulting in wrong measurements. An advantage of ultrasonic sensors is that they are not sensitive for sunlight that they are able to detect transparent objects such as windows.

Ultrasonic range finders have their limits just as every other sensor. Some of these limits are presented in a paper from Borenstein & Koren (1988). One of the limitations is caused by the angle of incidence. When the obstacle is inclined most of the sound energy will be reflected into a different direction, as can be seen in Figure 4-5. Borenstein concludes that a maximum inclination for a reliable detection of a smooth surface is only 25°.

The amount of reflected sound energy also depends on the texture of the obstacle. To obtain a highly diffusive reflection the texture of an obstacle should be comparable to the wavelength. As can be seen in Figure 4-6, (Kuttruff, 2009). Figure *a* represents the case in which the texture is smaller than the wavelength, *b* represent the case in which the texture width is similar to the wavelength and *c* represents the case when the texture is larger than the wavelength.

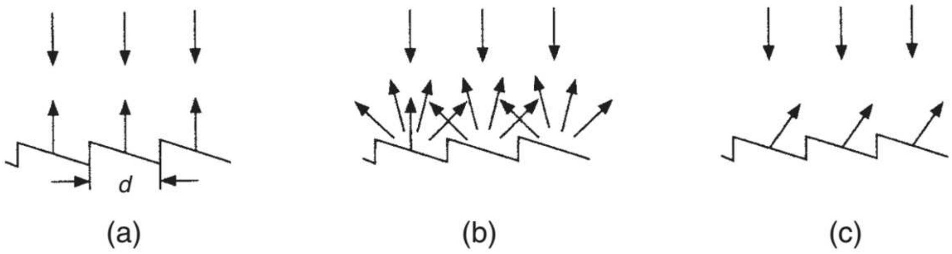


Figure 4-6: Scattering of a sound wave

$$\lambda = \frac{v}{f} \tag{4-7}$$

To calculate the size of the surface texture which causes a diffusive reflection Equation 4-7 can be used. In this equation λ represents the wavelength, v represents the speed of sound and f represents the emitted frequency of the sensor. When a sound of $50000Hz$ is transmitted the texture should be $6.8mm$ apart. The frequency could be increased but would require more power.

Another problem with reflection is that inclined reflection at a short distance can have a power reflection that is equal to an object which is perpendicular but at a larger distance. This problem can be minimized by improving the directionality, making the Field Of View (FOV) of each sensor smaller. However, more sensor are needed for the same coverage.

To solve the problem of interference between robots Borenstein suggest to use a different frequency for each robot. Interference could still occur between multiple ultrasonic sensors on the same robot. To solve this problem a different time delay for each sensor can be placed between each emitting/listening phase. In this way the signals from different sensors can be distinguished.

Another measurement error in ultrasonic systems is caused by the difference in the speed of sound. This problem is addressed by Müller et al. (2014), which mitigates the problem by calibrating the sensor for different temperatures. Another problem addressed by Müller is the effect of two environment conditions: the distance of the obstacle and the propeller noise. Müller uses a wall as obstacle to perform his measurements. The propeller noise is increased by increasing the thrust level. The performance of the sensor is expressed in the mean error and the standard deviation.

Implementation Besides the implementation by Müller et al. (2014) other implementation can be seen in the literature as well. An example in which multiple ultrasonic sensors are used is given by Holenstein & Badreddin (1991). In his work 24 ultrasonic sensors are placed in a circle around the robot. Also in the work of Koren & Borenstein (1991) such a ring is used.

Ultrasonic sensors are also used in the work from Petillot et al. (2001), sonar is used to detect and track underwater obstacles. Uncertainties in the measurements are taken into account using a Kalman filter.

4-4 Laser

Principle Another widely used option for measuring distances are laser range finders. Laser range finders are active sensors which can give accurate measurements, but generally have a high weight. Because of its higher accuracy it still is a viable options for large UAVs that can generate enough lift. But for small UAVs this becomes a problem (Jongho & Youdan, 2012).

To get an idea of the available techniques one can look at a paper from Amann et al. (2001). Amann puts the available optical distance measurement methods into three categories: interferometry, time-of-flight and triangulation. From these methods only time-of-flight methods are used in airborne applications, therefore interferometry and triangulation methods will not be discussed here.

Time-of-flight methods can be split into 'pulsed methods' and 'phase-shift' methods which are explained in Thiel & Wehr (2004). Both are active methods. Pulsed methods send a laser pulse and measure the reflection similar to sonar and active infrared. In phase-shift methods a continuous laser wave is send. The intensity of the laser is modulated with a well defined function such as a sinusoidal function. The time-of-flight is determined by measuring the phase difference between the transmitted and received intensity.

Performance For time-of-flight systems, the accuracy of the distance measurement depends on how accurate the time difference can be calculated. To obtain $1mm$

accuracy, the accuracy of the time interval measurement should be 6.7 ps.

An extensive research to evaluate the performance of laser range finders is done by Reshetyuk (2006). Reshetyuk classifies internal and external errors. Internal errors can be split into fundamental errors and errors specific to the hardware. Examples of internal errors are jitter and walk.

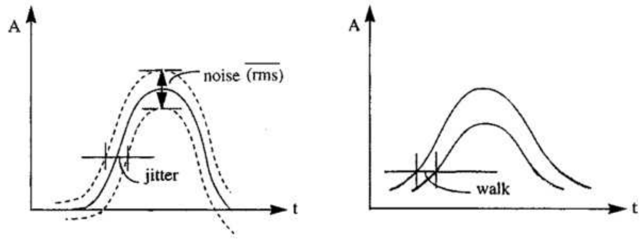


Figure 4-7: Jitter and Walk

Jitter and walk are shown in Figure 4-7. Jitter is caused by noise in the system due to background radiation or due to the electronics. This noise can cause variation in the amplitude, which has an effect on the time measurement. Also the distance has an effect on the precision. The amplitude decreases proportional to the square of the distance resulting in the so called walk error.

External errors can be split between environment effects and object related errors. Examples of errors caused by the environment are errors caused by humidity or air temperature. Object-related errors are related to the properties of the object such as the transparency or reflectivity. These classification can cause two types of errors random errors and systematic errors. An example of a random error is jitter, a systematic error is walk.

In another paper from Viswanathan et al. (2007) no fundamental limits are discussed but a comparison is being made between an active 3D time-of-flight infrared laser range sensor and a $4mm$ Bumblebee 3D stereo vision camera. The two sensors are compared from an OA perspective. The sensors are placed on a robot after which runs are performed to determine to following metrics: Correct Rejects(True Negative (TN))/Hits(True Positive (TP))/Misses(False Negative (FN))/False Alarms(False Positive (FP)). These are shown in Table 4-2 Also the mean stopping distances are compared.

The laser range finder and stereo vision show similar results. According to Viswanathan stereo vision is preferred because it is cheaper and versatile while having modest power requirements.

Implementation Laser range finders are used for obstacle avoidance in Bachrach et al. (2010). In his work a laser range finder is used to estimate the vehicle's motion and build a high resolution map. In work from Ficuciello et al. (2013) lasers are used to determine

Table 4-2: Comparison between stereo vision and infra-red

Test Condition	Hits		Misses		False Alarms		Correct Rejects	
	IR	*SV*	*IR*	*SV*	*IR*	*SV*	*IR*	*SV*
No Object	0	0	0	0	0	0	20	20
Wall	20	18	0	2	0	0	0	0
Walker	20	20	0	0	0	0	0	0
Cane	15	18	5	2	0	0	0	0
Person Stand	20	20	0	0	0	0	0	0
Person Walk	20	20	0	0	0	0	0	0
Totals	*95*	*96*	*5*	*4*	*0*	*0*	*20*	*20*

the distance to cars and pedestrians. The performance of the laser is determined, by measuring the distance in a controlled environment. The error of the laser used by Ficuciello stays under $30cm$. In the work from Ferrick et al. (2012) a laser range finder is used on an ARDrone. No detection performance is measured, but it shows laser range finders are possible for UAVs of comparable size. Also in the work of Hrabar (2011) and Delin et al. (2012) laser range finders are used, but without a detailed description of the system.

4-5 Radar

principle To complete the detection survey a brief evaluation of radar systems is given. Radar is a widely used method for detecting obstacles. Many research has been done to determine the detection performance of radar systems. This knowledge can for example be used in stealth fighters who need to prevent detection by enemy radar. Radar works on the same time-of-flight principle as infrared or lasers, the main difference is the frequency of the electromagnetic waves.

performance The range of a radar system is given by the radar equation given in Equation 4-8, in which P_S represents the transmitted signal, P_E the received power, G the antenna gain, λ the wavelength and σ the Radar Cross Section (RCS).

$$R = \sqrt[4]{\frac{P_S \cdot G^2 \cdot \lambda^2 \cdot \sigma}{P_E(4\pi)^3}} \tag{4-8}$$

The RCS is a property of the object and summarizes the reflectivity. Not only the radar equation defines the range of the radar, but also the waiting time. The upper limit determined by the waiting time can be calculated using equation 4-9. The time is represented by τ and the speed of light is represented by c. The minimum range of the radar depends on the transmitting time and can be calculated by replacing τ with the transmitting time in equation 4-9.

$$R = \frac{c \cdot \tau}{2} \tag{4-9}$$

The minimum distance is also related to the range resolution. To distinguish two obstacles the distance between them should be larger than the minimum range. The angular resolution depends on the beam width and is represented in Equation 4-10. In this equation S_A is the angular resolution, R is the range and Θ is the beam width.

$$S_A > 2R \cdot \sin\frac{\Theta}{2} \tag{4-10}$$

Equation 4-8 shows the range is dependent on the RCS(σ) of the obstacle. The goal of stealth fighters is to make the RCS as low as possible. The RCS will change depending on the orientation of the obstacle with respect to the radar. For specific cases such as flying planes these changes can be captured in a statistical model. To specify the statistical properties of the RCS of complex objects the Swerling methods where introduced by Peter Swerling. Five target models exist, As an example the first Swerling model is given in Equation 4-11. In this equation the probability of the RCS(σ) is given as a function of the average RCS, σ_{av}.

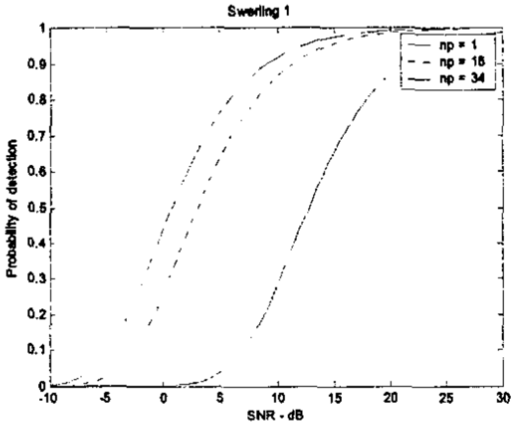
$$p(\sigma) = \frac{1}{\sigma_{av}} e^{-\frac{\sigma}{\sigma_{av}}} \tag{4-11}$$

The swerling models are also used in a research by Kwag in which radar systems are used in a Collision Avoidance System (CAS) for low-altitude UAVs. (Y. Kwag & Kang, 2004), (Y. K. Kwag et al., 2007). A result of the research is presented in Figure 4-8. The detection probability is shown as a function of the Signal to Noise Ratio (SNR). In this figure the first Swerling method is used and a false alarm probability of 10^{-6} is assumed. In the figure the increase in probability between the amount of pulses(np) can be observed. These kind of curves can be determined for a type of system and environment to determine the performance of an obstacle avoidance system.

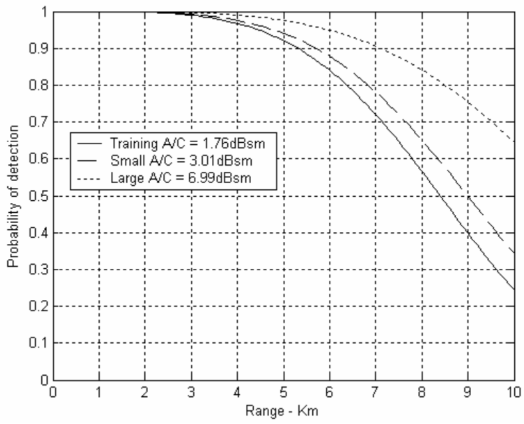
In Figure 4-8 b the relation between the range and the probability of detection is shown for three type of aircraft.

Again looking at Equation 4-8 one can see that also the wavelength has influence on the range. In a paper from Kemkemian & Nouvel-Fiani (2009), several frequencies bands are compared based on coverage, angular accuracy but also weight and cost are taken into account.

Implementation Also for radar systems several implementations are published. In the paper from Ariyur et al. (2005) a low resolution radar is used to avoid obstacles with a UAV. In work from Kandil et al. (2010) a radar sensor width a range from 0.2 to 30 meters is used with an angular coverage of 80 degrees. Unfortunately no extensive performance evaluation of the used radar system is presented.



(a) Probability to SNR



(b) Probability to Range

Figure 4-8: Detection probability with respect to SNR and range.

4-6 Combination of sensors

In the previous sections vision, infrared, sonar, laser and radar where discussed independently. Often UAVs are equipped with multiple sensors, the weakness of one sensor could be mitigated with a different sensor. An example is given in Fasano et al. (2006), which uses a radar system, infrared cameras and two vision cameras to avoid collisions. Also in the work of Matthies et al. (1998) combinations such as, stereo vision with CCD cameras and stereo with with InSB FLIR cameras, are implemented and compared. Matthies does not use radar and sonar sensors because of their significantly lower angular resolution.

4-7 Performance evaluation of detection

In the previous sections the available detection methods were discussed. Using this overview sub-question 1.2 can be answered. First the relevant performance metrics seen in the literature will be discussed after which the environment metrics are discussed.

Performance metrics To measure the performance some metric needs to be defined. The performance metrics seen in literature to quantify this consisted mainly of the RMS error of the distance measurement. In some cases also the variance and covariance are taken into account. Another metric is the Receiver Operating Characteristic (ROC) curve in which true positives and false positives are plotted. As a third metric the computational effort has been taken into consideration, for which the metric of Frames Per Second (FPS) or computation time was used. Only for stereo vision other metrics were proposed such as the percentage of disparities which are larger than a certain

Table 4-3: Parameters which influence the total error, specified for each sensor

		Monocular vision	Stereo Vision	Infrared	Sonar	Laser	Radar
Internal Errors	Fundamental	Flow assumptions	Baseline Focal length Min Disparity	Waiting time Transmitting time Power Optical crosstalk	Wavelength Waiting time Transmitting time Power	Walk Waiting time Transmitting time Power	Waiting time Transmitting time Beam width Power
	Hardware specific	Computational Power Resolution Lens Shutter time Internal noise	Computational power Resolution Lens Shutter time Internal noise	Transmission losses Receiving losses Internal noise	Transmission losses Receiving losses Internal noise	Jitter Transmission losses Receiving losses Internal noise	Transmission losses Receiving losses Internal noise
External Errors	Object	Distance Specularities Transparency	Distance Texture Contrast Texture angle Repetitiveness Specularities Transparency	Distance Speed Inclination Reflectivity	Distance Inclination	Distance Speed Transparency Reflectivity	Distance Speed RCS
	Environment	Illumination External noise	Illumination External noise	Temperature External noise	Multi-path Temperature Propeller Noise External noise	Temperature Humidity Multi-path External noise	Multi-path external noise

threshold. So performance metrics are RMS error, refresh rate and ROC curves. Since it is time consuming to measure the performance metrics in all environmental conditions it needs to be determined which environments are relevant.

Environment metrics To determine which environment metrics are relevant for assessing the detection performance it needs to be known what influences the performance metrics. Several causes are mentioned under the 'performance' paragraphs in the previous section. To summarize these effects the classification used by Reshetyuk (2006), which was mentioned in subsection 4-4, is used for each type of sensor. In this classification a distinction is made between internal errors and external errors. The internal errors are split between fundamental errors, which are caused by the physical principle, and hardware specific errors. The external errors are split between object related errors and environment errors. An overview of the parameters which influence these errors is given in Table 4-3.

In Table 4-3 it can be seen that stereo vision has some fundamental limits due to the base line and the focal length of the camera. For sonar the transmitted frequency determines the maximum size between irregularities which can be detected. For radar it was shown that the angular resolution depends on the beam width and the range resolution depends on the transmitting time τ.

The external factors are also shown in Table 4-3. For stereo vision for example the texture and illumination are important. Texture and illumination have several aspects. For texture not only the amount of texture but also the angle and repetitiveness is important. Other light effects such as specularities and shadows have an effect on the detection error as well. Also transparencies influence the detectability of an obstacle. Lasers, for example, are not able to detect fully transparent objects such as windows. For active systems the orientation of the object has an influence on the detection performance. For sonar it was stated that a smooth obstacle is detectable until it has an inclination of about 25°. Related to this inclination is the shape of the overall object. For radar such characteristics are summarized in the radar cross section. The Swerling models can be used to simulate the statistical properties of such objects. Unfortunately no equivalent model exist for other sensors. Some properties of the environment influence the performance of all sensors. Such as distance and external noise.

Sub-question 1.2 can be answered by looking at Table 4-3. Each environment which influences the error is a relevant environment metric to asses the performance of the sensor. For each sensor a different set of environment metrics is relevant and therefore the different environment metrics stated in the table can be used to define the performance tests for each sensor.

Chapter 5

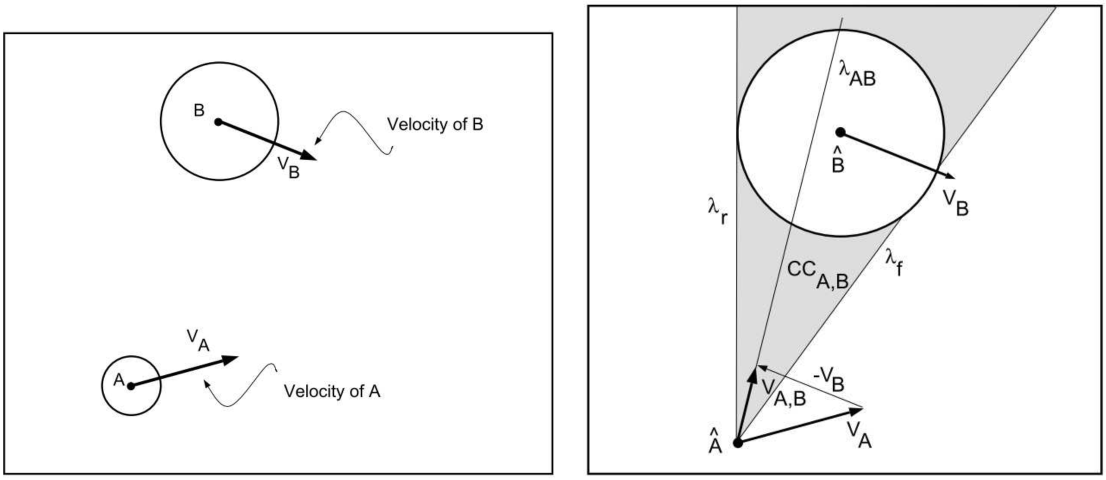
Avoidance maneuver

The second part of an OA system determines the avoidance maneuver and consist of two parts: conflict detection and avoidance. Both are discussed in the following sections.

5-1 Conflict detection

The next step in an OA system after detection is conflict detection. Not every OA system uses conflict detection, which can also be seen in Table V. The first column of the section conflict detection specifies if such a method is present. A conflict detection method propagates the state of the UAV into the future, after which a check is done if the propagated state is in conflict with the detected obstacles. State propagation is mainly done in three ways. The state could be propagated linearly, in which it is assumed the UAV will keep flying straight. Also a flight plan/flight path can be used to recognize conflicts. The third approach is 'worst case' propagation in which all possible flight scenario's are taken into account. Other characteristics of conflict detection stated in Table V are: UAV dynamics and Obstacle uncertainty. Both can be taking into account when detecting a conflict. Finally a distinction is made between binary conflict detection in which a Boolean (collision/no collision) is calculated and conflict detection in which a set of safe and unsafe trajectories is calculated.

An example of linear propagation is given by Fiorini & Shiller (1998), in which a Collision Cone (CC) is constructed to detect a collision. The collision cone can be seen in Figure 5-1. In the figure, $V_{a,b}$ represents the relative velocity between $\hat{A}$ and $\hat{B}$. By drawing lines from $\hat{A}$ tangent to the circle $\hat{B}$, the collision cone is constructed. When the relative velocity $V_{a,b}$ lies inside the collision cone a conflict is present. The radius of $\hat{B}$ depends on the obstacle sizes of A and B, which are shown in Figure 5-1a. Often a safety region

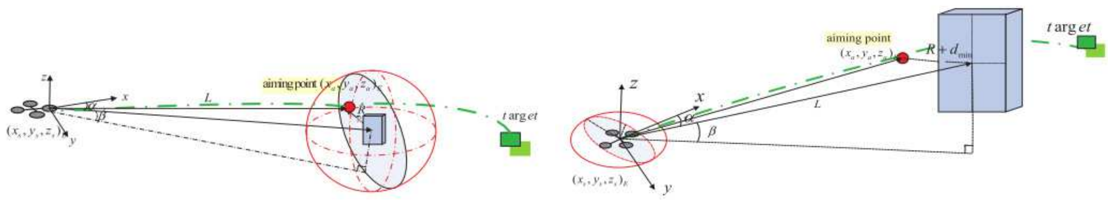


(a) **(b)**

Figure 5-1: Visualization of the collision cone

is added as well. Similar approaches are done by Jongho & Youdan (2012), Watanabe et al. (2007) and Alejo et al. (2014).

The safety region can be defined around the UAV or the obstacle. A comparison between the two methods is done by Geng et al. (2013). The two methods are called safety-ball model and mass point model, which can be seen in Figure 5-2. Geng uses the collision detection to construct an 'aiming point' which serves as a temporary waypoint to avoid the obstacle. This close relation between collision detection and avoidance. The performance of the two methods is based on the length of the trajectory from which Geng concludes that the 'safety ball model' is the preferred method since the length of the avoidance maneuver is smaller. This is reasonable since to fit a large obstacle into a sphere, the sphere itself needs to be large. Which results in a larger safety zone for large obstacles.



(a) **(b)**

Figure 5-2: Mass Point Model (left) and Safety Ball Model (right)

In a paper from Hrabar (2011) an elliptical shape is fit around the obstacle and a circle safety region around the UAV. This circle is aligned perpendicular to the flight direction which is propagated linearly. This creates a cylindrical safety volume for the UAV from which a collision is checked with the elliptical regions of the obstacles.

A method in which the 'worst-case' propagation is nicely illustrated is a method by Frew & Sengupta (2004). The geometric regions defined by Frew can be seen in Figure 5-3. The yellow rectangle is the obstacle. The purple region is a set of all possible states that the system can enter from a given state. The blue region is the set of all states which lead to collision. To account for measurement errors a certain safety margin is added to the blue region to create the light blue region called the 'unsafe set'. The 'worst-case' obstacle detection is shown as the green rectangle in the figure. When the UAV is in the unsafe set a collision will occur.

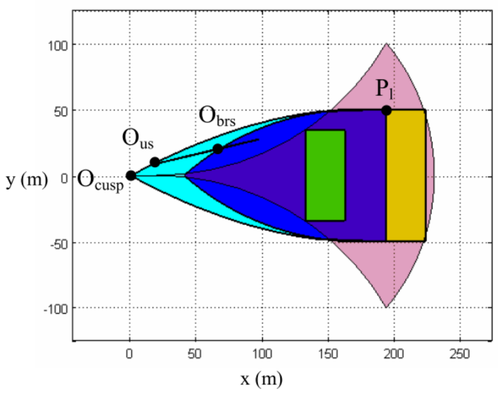


Figure 5-3: Obstacle (yellow), Detected obstacle (green), Forward reachable set (purple), Backward reachable set (Blue), Safety region (Light blue)

Finally a method in which a flight path is used to detect collisions is presented by Shah & Aouf (2009). Shah uses a simple safety distance, if the distance of the flight path with respect to the obstacle is closer than the safety distance a collision is registered.

5-2 Avoidance

After an obstacle or collision is detected an avoidance maneuver needs to be determined, this can be done in many ways. In this literature study a division is made based on the most seen avoidance methods in Appendix V. The following categories will be discussed: behavior based methods, knowledge based methods, vector field methods and path planning. Another conclusion from Appendix V can be made which is that most papers do not discuss the performance of the avoidance method. Therefore only the principle and implementation will be discussed for each category.

5-2-1 Behaviour based methods

Principle The first method to be discussed are behavior based methods. In behavior based methods the sensor identifies a type of environment or condition after which a certain behavior is selected. These methods include the simplest form of obstacle avoidance methods in which two scenarios are present: with obstacle and without an obstacle. An example would be to apply a change in heading (with a predefined value) when an obstacle is present. The advantage of this system is that the computational cost is extremely low. Such a method however might not be very efficient. Traveling time and distance might be larger when applying a simple rule, which can be troublesome when the power supply is limited.

Often the combination between collision detection and behavior based methods is seen. The collision detection gives a flag or a set of unsafe trajectories. When an unsafe set is known also a safe set is known. Often a certain rule is applied to select one of the safe options. Instead of using a rule to select the velocity vector also a way-point or escape point can be constructed. This can be done by selecting a point on the edge between the safe and unsafe set. The latter is sometimes referred to as geometric resolution.

Implementation An implementation of a behavior based method is done in work from Moufid et al. (2008). Harb detects seven types of environments: corridor, cross roads, frontal T shaped cross-road, left T shaped junction, right T shaped junction, left corner and right corner. Based on the detected environment a certain control behavior is selected such as Keep Right, Pass Cross or Keep Left.

A similar approach is used by Schafer et al. (2005) in which laser range finders and stereo systems are used to select behaviors such as: KeepDistLeft, KeepDistRight, Evasion or Obstacle Stop. Another example is given by Delin et al. (2012). In his paper the behaviors: Hovering, Turning, Forward Flight and Gate Exploration are used to navigate through an obstacle field. The size of the obstacle-free section determines which mode is activated. Also in the work of Müller et al. (2014) a behavior based method is used. Müller makes a distinction between three environments: No wall, one wall and two walls. If there is no wall present the UAV will hover in its current position. When one wall is present the distance to the wall is kept constant. With two walls the distance to each detected wall is equalized.

Another example is given by Fiorini & Shiller (1998). Fiorini creates a set of safe velocities. To determine which of these velocities is selected three rules are proposed; highest velocity in line to its goal, maximum velocity within a specified angle α or the velocity with a minimal risk. Depending on the situation one of the rules is used. A rule is also applied by Alejo et al. (2014) in which the velocity closest to the desired velocity is chosen.

5-2-2 Knowledge based methods

Principle The second type of avoidance method is called knowledge based. Knowledge based methods are a set of methods which use a 'knowledge base' to solve the problem. It can be used to solve a diversity of problems. An example of such a method is fuzzy based control. In fuzzy based control multiple if/then rules are constructed after which fuzzy sets are defined to map the input to the output.

Another knowledge based method is a neural network. Neural networks are a (simplified) mathematical model of the brain. It uses 'neurons' to perform a mapping between inputs and outputs. These neurons can be 'trained' to create the required behavior. The training can be done online or offline.

A third knowledge based method is reinforcement learning. In reinforcement learning a specific policy is learned. For each state the optimal control action is determined, this is done by making using of a reward function. This reward function is dependent on the interaction with the environment.

Implementation An example of a fuzzy control implementation is given by Dong et al. (2005). The inputs to the system are the distance and angle to an obstacle, the output is an angular acceleration. Also in the work form Darío et al. (2014) a fuzzy controller is used to define the control reference.

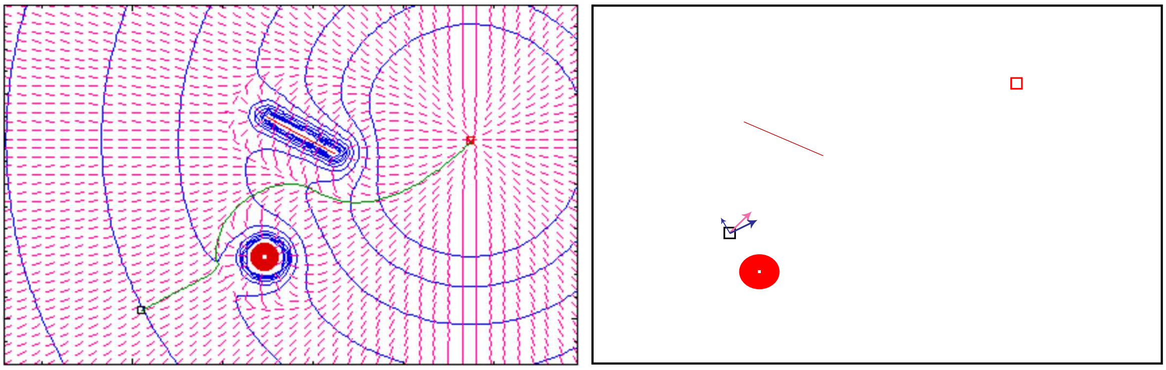
An example of reinforced learning is presented by Ross et al. (2013), the policy is trained by comparing the control policy with the control given by a pilot. In such a way a strategy is created which is able to successfully avoid obstacles.

5-2-3 Vector field methods

Principle One of the most popular avoidance methods are vector or potential fields methods. The potential field principle is particularly attractive because of its elegance and simplicity (Koren & Borenstein, 1991). Detected obstacles result in a repulsive force or moment as shown in Figure 5-4a. The sum of the forces or potential fields results in a certain control output.

In a paper from Tilove (1990) an overview of the field is given. As stated in the paper there are two basic approaches: global and local approaches. In the global approach a map of the environment is made. In this map each coordinate is assigned a vector(or a certain potential). The control action is found by searching an optimum path in this map. In the local approach no map is being made but the vectors are calculated using the current measurements only.

The forces or potential fields can be defined in several ways. Again two options are described by Tilove: classical and generalized ways. In the classical option the force or potential only depends on the position of the robot. Which means that a large repulsive



(a) (b)

Figure 5-4: Global (left) and local (right) force field methods

force of a nearby obstacle is present even if the velocity vector is not pointing in that direction. The generalized potential does take the velocity vector into account.

When the potential field is calculated, a control action needs to be determined. Tilove describes two basic methods to convert the field into a control input: hill climbing and force control. In hill climbing the robot simply moves in the direction of the resultant force or in the direction of the steepest descent. In the other method the force is treated as a physical force acting on the agent resulting in an acceleration. This can be an angular acceleration or a translational acceleration.

Tilove concludes that a generalized potential field should not be used with hill climbing. He also states that the biggest downside of force field methods are local minima. When the resultant force is zero the robot will stay in the same position, but also oscillations might occur due to the way in which the control and field function interact. Luckily there are solutions present to solve these kind of issues. To prevent oscillation, damping terms could be added into the control function, but also more advanced methods have been developed.

The problems occurring in potential field methods are also discussed by (Koren & Borenstein, 1991), in which the Virtual Force Field (VFF) method is analyzed. The VFF is a global method which uses a 2D grid and a confidence value on each point of this grid to represent the probability of a point being an obstacle. It uses force control to determine the output control. Koren states the following problems:

- Trap situations due to local minima
- No passage between closely spaced obstacles
- Oscillations in the presence of obstacles
- Oscillations in narrow passages

The problems are the same as the ones stated by Tilove, but Koren comes with a solution which is stated at the end of his paper:

For these reasons we have abandoned potential field methods altogether, and developed a new method for fast obstacle avoidance. This method, called the Vector Field Histogram (VFH) method produces smooth, non-oscillatory motion.

The VFH method is explained in (Borenstein & Koren, 1991). The VFH method employs a two-stage data-reduction technique instead of one. The Cartesian grid used in the virtual force field is transformed into a one dimensional polar histogram. In the histogram 'valleys' are found, these valleys are used to steer the robot. When a robot is approaching an obstacle head on, the velocity is reduced.

Another method to solve the problem of local minima is stated by Park et al. (2001). Park solves the problem by making use of simulated annealing. In simulated annealing a new direction is chosen randomly, this direction will be chosen if the potential is lower then its current potential. If this is not the case the new direction still might be accepted based on a probability of $e^{-\frac{\Delta}{T}}$. In which T represent the temperature, when the temperature is high the probability a higher potential will be accepted is higher. The simulated annealing is activated when a system is in a local minima. It starts with a high temperature after which it decreases with a certain cooling rate .

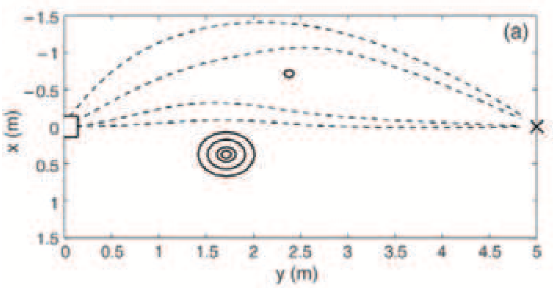


Figure 5-5: Simulated paths from Huang et al.

Implementation In Appendix V six implementations of potential field methods can be seen. Chen et al. (2013) implements a vector field method by making use of Lyapunov functions. To determine its performance Chen compares the desired trajectory to the actual trajectory. In another paper written by Marlow & Langelaan (2011) a local occupancy grid is used. It uses a generalized function and a 'hill climbing' approach to find the control output.

Huang et al. (2006) uses a reactive method and force control to steer the agent. Its performance is tested on a specific obstacle course which is successfully avoided. Also the length of the avoidance maneuver is compared between several tuning variables which can be seen in Figure 5-5

In a paper from Stowers et al. (2011) no direct performance is measured, but the output of the algorithm is shown. The last paper mentioned in the table is a work from Kandil et al. (2010). Kandil uses a reactive general method which uses force control to calculate a moment that is applied to a helicopter. Unfortunately also in this paper no performance is being measured.

5-2-4 Path Planning

Principle Another method often used for creating avoidance maneuvers is path planning. Many approaches exist to solve the path planning problem. A distinction can be made between local and global path planning. Which is explained by Mujumdar & Padhi (2011). In global path planning the total path from current position to the goal is created. In local planning only part of this path is generated.

Several global path planners are mentioned by Mujumdar. The first method mentioned are so called Graph Search Methods, which search for a feasible path in the given environment. Methods such as A*, D*, Voronoi graphs or probabilistic road-maps can be used. The A* method finds the optimal path by expanding the path which has the lowest cost function. This cost function consist of a path-cost function and a "heuristic estimate " of the future path cost-function. The search is ended when the goal is reached and no other expansions lower the cost function. Other graph searches use similar approaches.

Another method described by Mujumdar is the Rapidly exploring Random Tree (RRT) method. In RRT a random sample of the environment is generated at each step. The node that lies closest to his point is selected to grow an incremental distance in the direction of the random point. If this new node is feasible (collision-free and reachable) the node is accepted. The iteration continuous until the goal is reached.

The disadvantage of graph search methods and RRT methods is that they require a lot of memory and computational power. Also the RRT solution is not optimal. To lower the computational cost, the map used by the algorithm could be represented as quadtrees (Mujumdar & Padhi, 2011), which is a data structure to represent the environment efficiently. In this representation the environment is divided unto quadrants until a quadrant is filled with the obstacle.

As a pure local method Mujumdar mentions Model Predictive Control (MPC). Again a big research field. Model predictive control performs online optimization over a finite receding horizon. Collision avoidance is built into the optimization problem by adding an extra cost term into the cost function. The MPC cost function is represented in Equation 5-1, which could be optimised using Dynamic Programming (DP).

$$L(x, u) = \frac{1}{2}(x_r - x)^T Q(x_r - x) + \frac{1}{2}(u_r)^T R(u_r) + S(x) + \Sigma_{l=1}^{O} P(x_v, n_l) \tag{5-1}$$

In this equation the first term is the state of the system(Q), the second term the control inputs(R), S represent the penalties due to state constraints and P represent the penalty of the obstacle. Another survey about path planning from Goerzen et al. (2010) was mentioned in section 3-3. In this survey similar approaches as the ones set out by Mujumdar & Padhi (2011) are discussed. The two survey papers give a good overview of the available techniques but both do not discuss specific implementation. For specific implementation one can again look at Appendix V.

Implementation One of the papers in the table written by Hrabar (2008) implements the D* lite graph search method. In this method an initial path is planned which is updated when an obstacle is detected. The performance of the algorithm was evaluated by the amount of runs that reached the target. For Hrabar this success-rate was 21 out of 27.

A method that was not discussed in detail in the previous discussion is implemented by Yang & Wenjie (2014). Yang uses a genetic algorithm. Genetic algorithms are randomized search algorithms based on natural selection. A grid is created in which each node has a value of one or zero. A node is one if it's part of the path and zero when it's not. By using a fitness function several paths are compared, the most 'fit' paths are used to create new paths. Two genetic algorithms are compared by Yang and are evaluated based on path optimality.

Model predictive control is implemented by Joelianto et al. (2013). In which the trajectory and control input are presented. No real flights are being performed and because of the abstract description of the obstacles it might be hard to implement the method in real life. In (Lee et al., 2011) a model predictive controller is used as well. The paper compares MPC with a regular PID controller. Lee concludes that the path of the model predictive control is more optimal. Another MPC implementation is done by Boivin et al. (2008), Boivon does not use a penalty in the cost function but avoids the obstacles by adding constraints to the optimization process.

5-3 Performance evaluation of the avoidance maneuver

So can the information presented in the previous sections be used to answer sub-question 2.2 from section 2-2. First the performance metrics are discussed after which the environment metrics will be addressed.

Performance metrics To determine the performance of the avoidance method again some performance metrics need to be defined. Metrics initially thought of were computational time, probability of collision and path optimality. These metrics were indeed found in the literature although most cases do not specifically determine the performance of the algorithm. Most papers plot the covered distance and additionally plot the states or the control input. When no obstacle is hit the conclusion is made that the algorithm works well. Some papers perform multiple runs and determine some sort of success rate.

Table 5-1: Parameters which influence performance

	Behavior Based	Knowledge Based	Vector fields	Path Planning
Fundamental	?	?	?	?
Environment	?	?	?	?

Another often seen metric is the separation distance, which can be seen as a path optimality measure. So to conclude three types of metrics are present: The success rate, path optimality and computational effort. The success rate consist of the percentage of successful flights. Unsuccessful flights can be split into flights which lead to a collision and flights which get stuck into a local minimum. There exist some hierarchy in the first two metrics since path optimality can only be determined for successful flights. Also from a design perspective the first goal of a system would be to prevent collisions after which the optimality of the path can be improved.

Environment metrics To determine which environment metrics are relevant for assessing the avoidance performance one could take a similar approach as was done for detection. For detection the environments for which the sensors are sensitive were selected and summarized in table 4-3. A similar table could be constructed for the avoidance methods. Unfortunately the fundamental and environmental parameters are not well known. These limits are barely discussed in literature and therefore Table 5-1 is filled with question marks. Instead of describing the limits researchers test the algorithm in a specific environment, these environments can still serve as inspiration for defining a set of environment metrics. The environments used by researchers are listed in appendix V.

In appendix 1-2 a distinction is made between real environments and simulated environments. For simulated environments two types can be observed. In the first type a large field of obstacles is being created. These can be predefined or created based on random processes. This difference is also illustrated in figure 3-3, in which an overview of the OA loop is given. In the figure it can be seen that the performance of the avoidance part(B), is dependent on the environment conditions B. These conditions can be fixed but also variable which is represented by "Δ Conditions B" and the red arrow.

Examples of these variable environments are the Markovian obstacle field from Sebesta & Baillieul (2012), fast-time Monte Carlo Simulation in (Kuchar & Kuchar, 2005) or the ergodic forest created in Karaman & Frazzoli (2012). The advantage of such methods is that the environment can be described using statistical metrics and a relation can be found between the performance of the obstacle avoidance system and these metrics.

The other type of simulation environment is when only one, maybe two objects are present. These objects are most of the time sphere or walls. Most of the papers only use one type of object, which does not say a lot about the performance of the system. These type of environments often lack a metric to describe the environment.

Also a subdivision can be made for real flight tests. First of all between indoors and outdoors. For indoor environments two types exists. One in which a UAV flies through a hallway and environments in which a specific obstacle course is set out. For outdoor environments also two options are present. Environments in which a UAV has to fly up and around buildings and forest environments. Unfortunately none of the papers uses a metric to describe the environment or a motivation why that specific environment is used. Also no structured changes in the environment are being made.

So from the environments seen in literature only the randomly generated simulated environment make use of specific environments metrics. Sebesta & Baillieul (2012) uses the width of the obstacle and the with of the open space between the obstacle. Which are practical environment metrics to determine the performance. In Kuchar & Kuchar (2005) a set of 30 random parameters is used to simulate the environment. Examples of these parameters are, heading, velocity and miss distance. Finally, Karaman & Frazzoli (2012) uses a 'tree-density' to describe the environment.

These metrics and the environments seen in literature were used to come up with a set of four environments metrics which are explained in the following paragraphs.

1. Length of the avoidance maneuver

The first environment metric specifies the length of the avoidance maneuver. Which is similar to the width of the obstacle used by Sebesta & Baillieul (2012), it could quantify the difference between the forest environment and the building environment discussed previously. The avoidance length can be specified by dividing the length of the path with obstacles by the length of the path without obstacles. Which is represented as the Avoidance Ratio (AR) in equation 5-2.

$$AR = \frac{P^*}{\hat{P}} \tag{5-2}$$

In equation 5-2, P^* represent the optimal path in the obstacle field and $\hat{P}$ represents the distance to the goal without obstacles. Another way to specify the length of the avoidance maneuver is by averaging the minimum distance to avoid obstacles, which is represented in equation 5-3

$$DCE = \frac{1}{n \cdot r} \int_0^n s_{escape} dt \tag{5-3}$$

In which Distance to Closest Escape (DCE) is the average of the escape distance. The escape distance is the distance between the edge of the projected obstacle (plus the UAV radius), and the intersection between the obstacle and the goal heading. Which can be seen in figure 5-6.

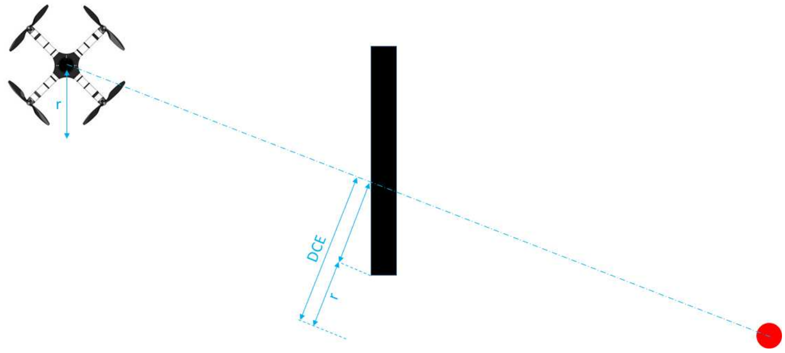


Figure 5-6: Distance to closest escape

2. Density of obstacles

The second chosen metric is the density of the obstacle. This density could be represented by dividing the amount of obstacles by the total space or surface (dependent if the test is for 3D or 2D applications). Another way to look at the density of obstacles is to measure the ease from which one could move from a to b. This can be quantified by selecting a random point and random direction in the flight space. For this point and direction the maximum distance which can be flown in a straight line can be determined. The average of these distances gives a measure of how densely packed the environment is. A third option is to only look at the closest distances between objects and average overall neighboring objects. Which is represented in Equation 5-4.

$$APD = \frac{1}{n \cdot r} \sum_{0}^{n} s_{passageway} \tag{5-4}$$

In Equation 5-4, the Average Pass-through Distance (APD) is given. In the equation n represent the amount of passages, r represent the radius and $s_{passageway}$ represents the closest distance between obstacles.

3. Percentage of dead-ends

The third metric is the percentage of dead-ends. Dead-ends especially increase the chance of getting stuck in a local minimum and therefore influence the overall performance of the algorithm.

$$DEF = \frac{1}{\pi r^2} \sum A_{deadends} \tag{5-5}$$

The percentage of dead-ends can be calculated using Equation 5-5. In this equation $A_{deadends}$ represents the total surface of positions in which a UAV could get trapped and r represents the radius of the UAV.

4. Freedom of movement

The fourth metric represents the freedom of movement. An OA task is 'easier' when a lot of space is available or in other words when many escape maneuvers are possible. An OA task becomes more difficult when conducted in a small room. Also an increased velocity lowers the amount of possible escape maneuvers. This difficulty can be quantified by adding the amount of states which cause certain collision. To get an idea on how to calculate such measure one could look at figure 5-7.

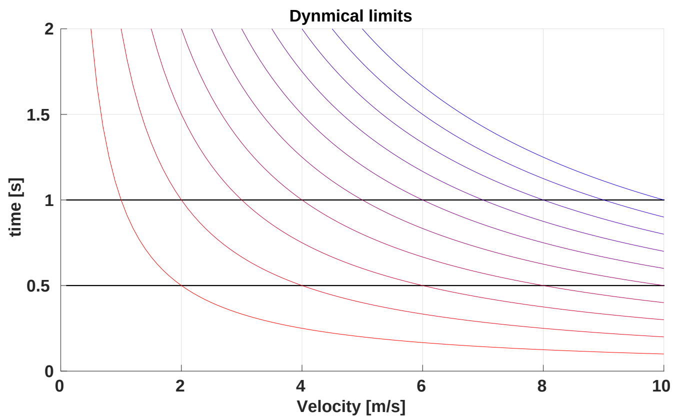


Figure 5-7: Dynamical limits

In this figure the time to collision is plotted as a function of the relative velocity between the UAV and the obstacle. The available time depends on the distance from which obstacles can be detected. For detection distances from $1m$(red) to $10m$(blue), the time to collision is plotted. The black lines represent specific time durations of an avoidance maneuver. If the time duration of the avoidance maneuver is larger than the time to collision a collision will occur. Also more advanced calculation could be done, for example the work of Karaman & Frazzoli (2012) shows that at certain velocity/density pairs a collision will certainly happen.

The amount of states at which a collision is certain depends on the dynamics of the UAV, the velocity and the overall delay in the system. So there is a lot of information hidden in this metric, but it could give a quantification on how difficult certain environments are.

(5. Curvature of the obstacles)

A fifth optional metric is the curvature of the obstacles. In such a way a distinction can made between circles and square sized obstacles. But it remains an open question if this really increases the difficulty of an OA algorithm. Therefore the focus will be put on the first four metrics.

Chapter 6

Conclusion

To develop an obstacle avoidance system for a serving application a method is needed to determine the performance of such systems. The goal has therefore been set to define, measure and compare the performance of obstacle avoidance systems. To come up with such a method a literature survey was conducted to get an overview of the obstacle avoidance field. Also the methods used to analyze these methods are elaborated.

Many types of obstacle avoidance exist which were summarized in a table. The types of obstacle avoidance were categorized on top-level aspects; Single/Multiple, Online Navigation/Separate CO and mapping/no-mapping. Next to that the methods where split based on the type of object it was able to avoid: Static/Dynamic and collaborative/non-collaborative.

The method itself is split between a detection part and a part which calculates the escape trajectory. Both parts were investigated to determine which performance and environment metrics are relevant to asses the performance of the algorithm.

The detection part is split between the type of sensors used. Vision, Infrared, Sonar, Laser and Radar. Each sensor had its advantages and disadvantages. Also the performance of each sensor is sensitive to different external characteristics. For vision the light conditions are very important, while for active sensor such as radar and sonar the inclination of the object is critical. The performance metric used for detection stayed restricted to distance errors, variances, ROC curves and update rates.

For the determination of the escape trajectory four methods where distinguished: behavior based, knowledge based, potential fields and path planning. Also for this part it can be said that each method has its advantages and disadvantages. Several implementations were evaluated, but most papers do not discuss the performance of the algorithm and lack a metric to describe the environment. Some papers have a statistical method to describe the environment. The performance metrics seen in literature were success rate, path optimality/(separation distance) and computational power.

Part IV

Additional Results

1 Overview test scenarios

Traversibility

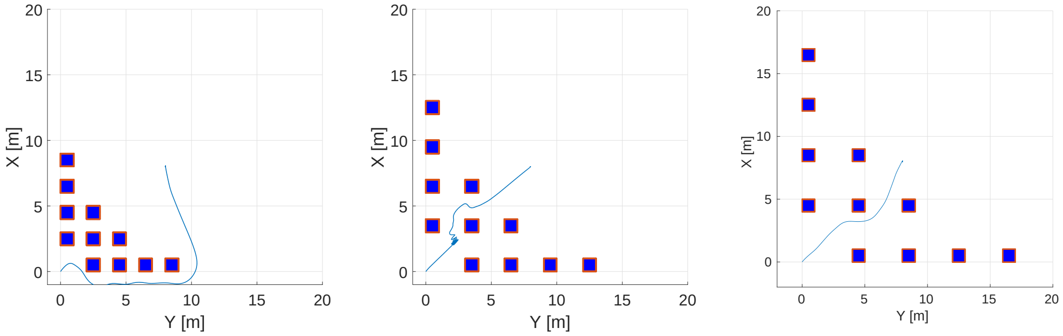


Figure 1-1: Traversibility tests

Collision State Percentage

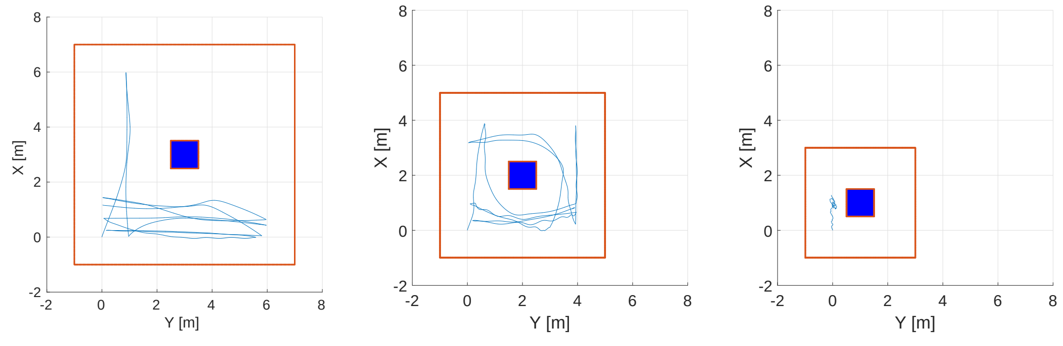


Figure 1-2: Collision state percentage tests

Average Avoidance Length

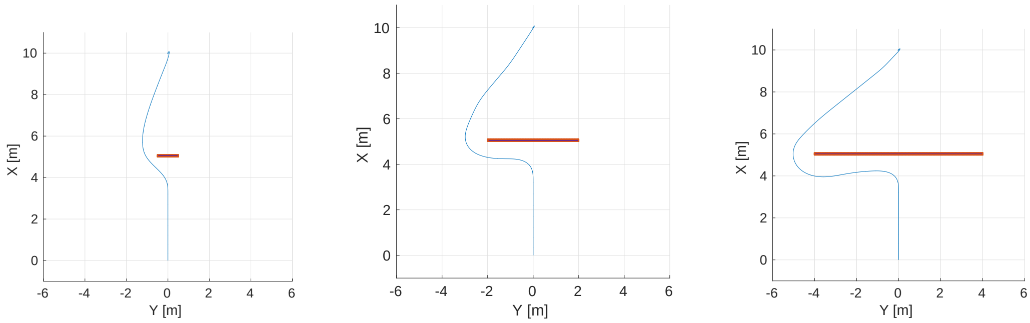


Figure 1-3: Average Avoidance Length tests

Dead-End Percentage

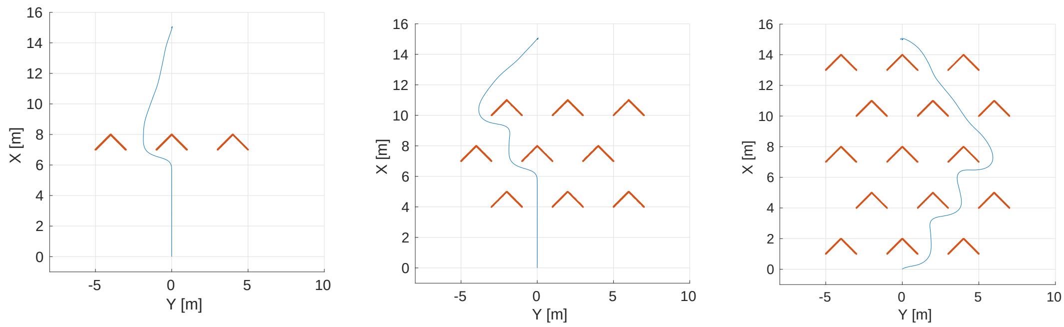


Figure 1-4: Dead-End Percentage tests

Average Orientation Angle

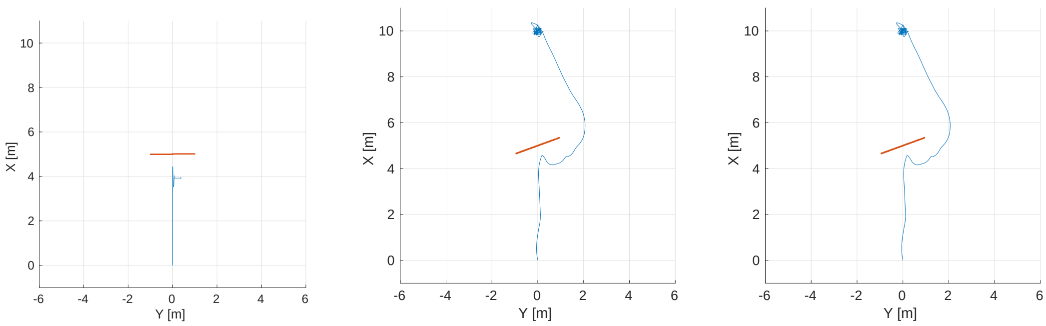


Figure 1-5: Average Orientation Angle tests

2 Results of the simulations and real-flight tests

Results of the 'Traversibility' tests

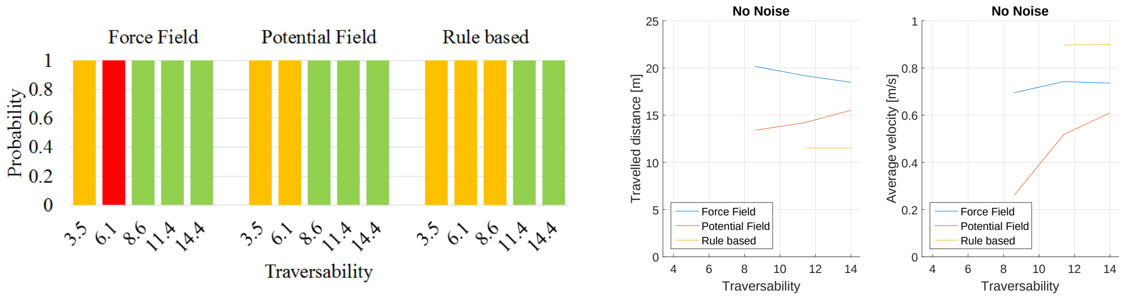


Figure 2-6: Results traversibility simulations without noise

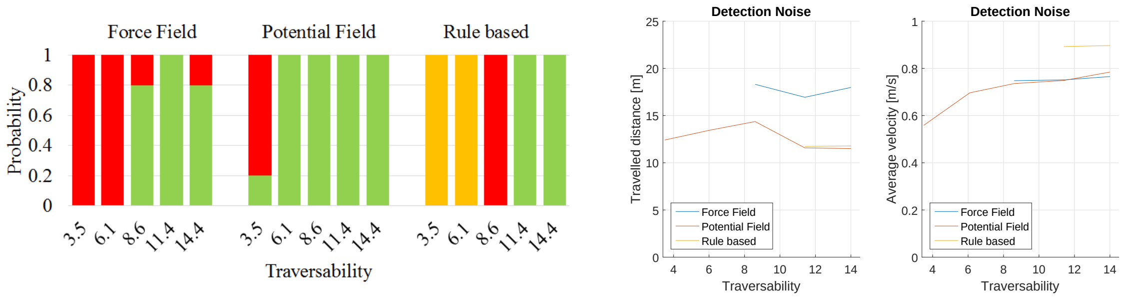


Figure 2-7: Results traversibility simulations with detection noise

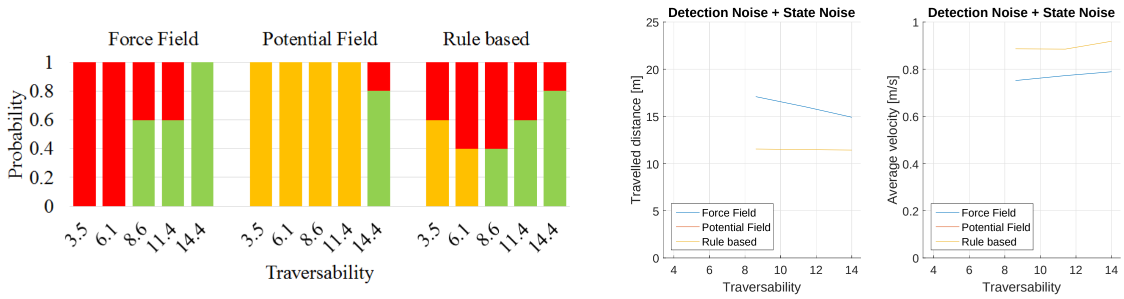


Figure 2-8: Results traversibility simulations with detection and state noise

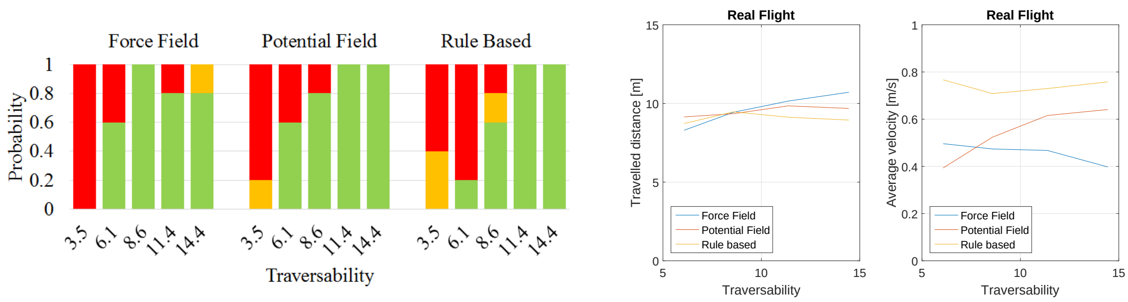


Figure 2-9: Results traversibility flight test

Results of the 'Collision State Percentage' tests

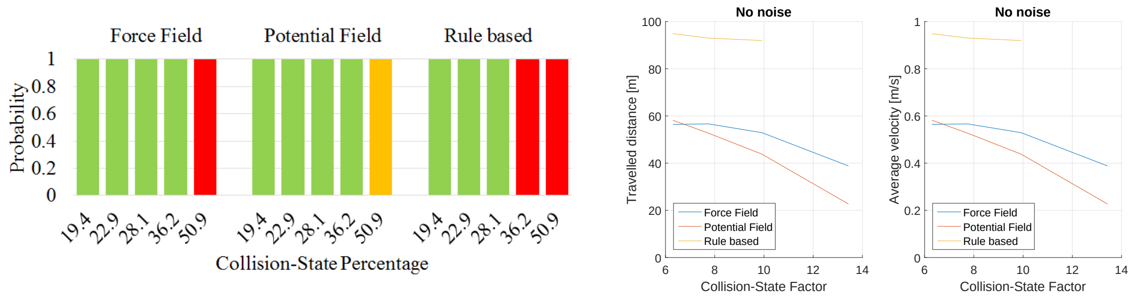


Figure 2-10: Results collision state percentage simulations without noise

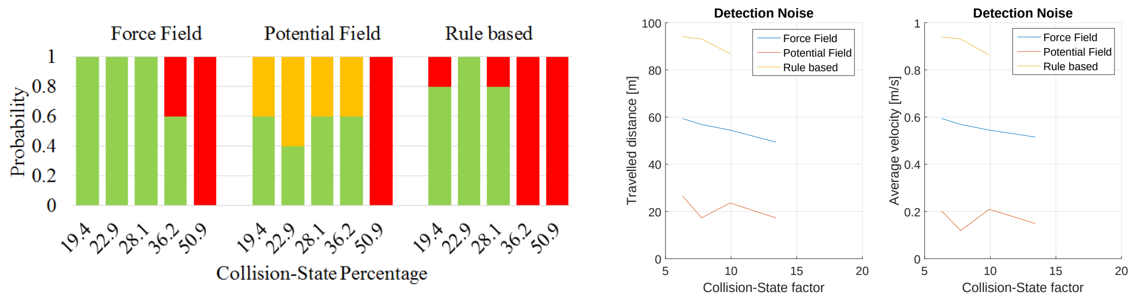


Figure 2-11: Results collision state percentage simulations with detection noise

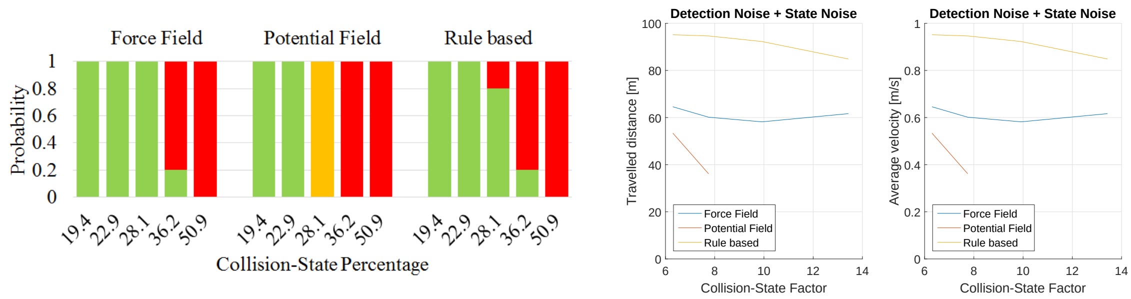


Figure 2-12: Results collision state percentage simulations with detection and state noise

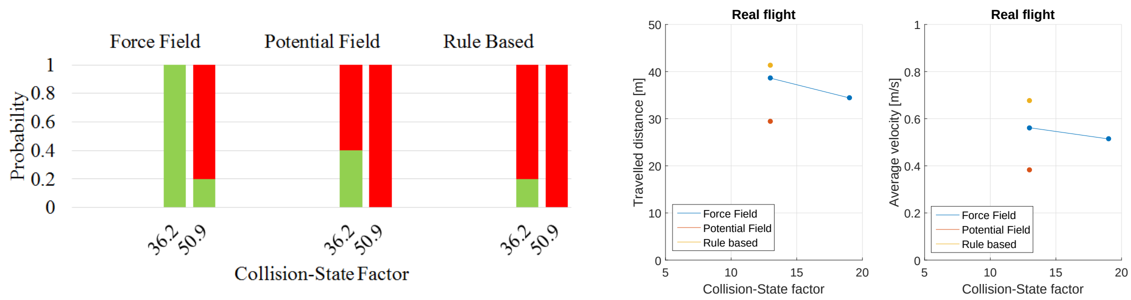


Figure 2-13: Results collision state percentage real-flight states

Results of the 'Average Avoidance Length' tests

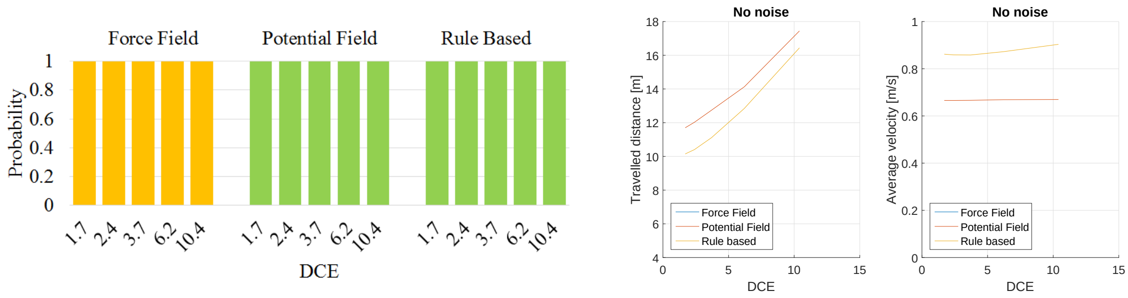


Figure 2-14: Results average avoidance length simulations without noise

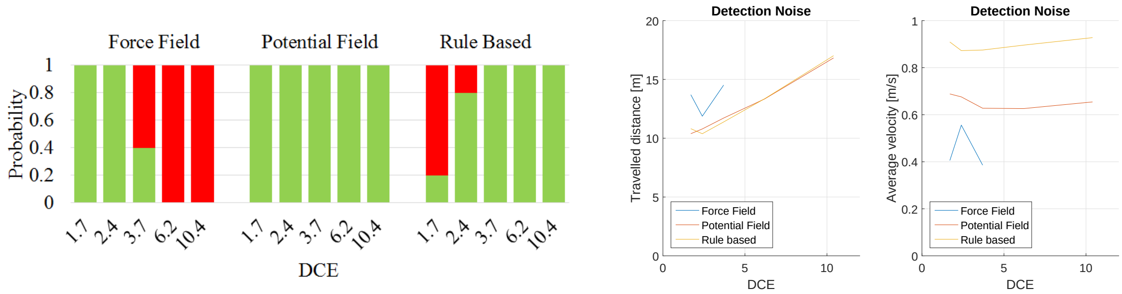


Figure 2-15: Results average avoidance length simulations with detection noise

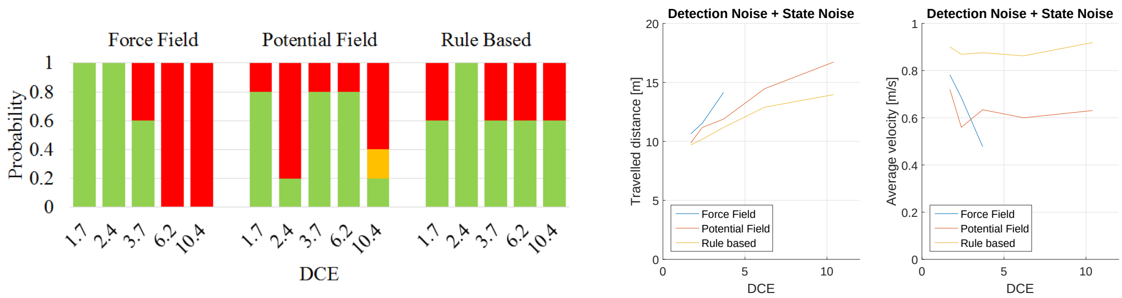


Figure 2-16: Results average avoidance length simulations with detection and state noise

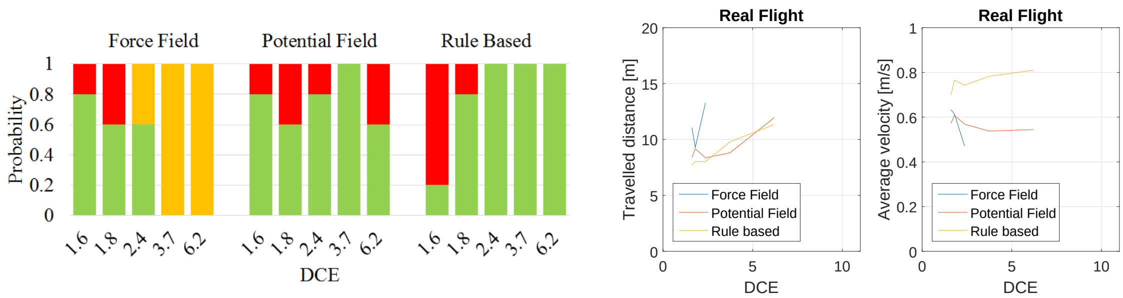


Figure 2-17: Results average avoidance length real-flight tests

Results of the 'Dead-End Percentage' tests

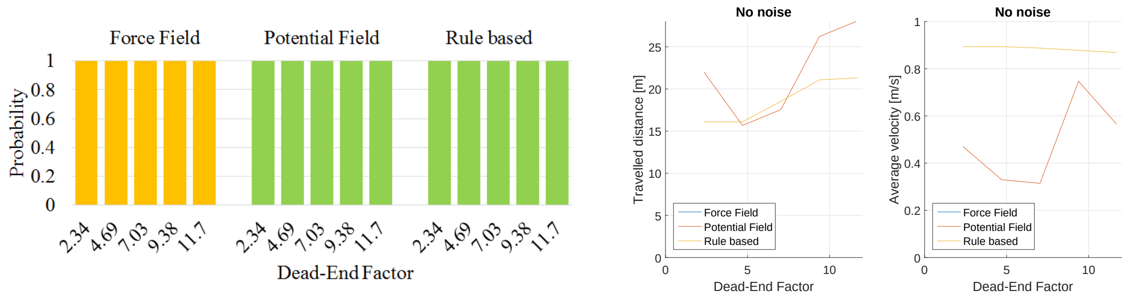


Figure 2-18: Results dead-end percentage simulations without noise

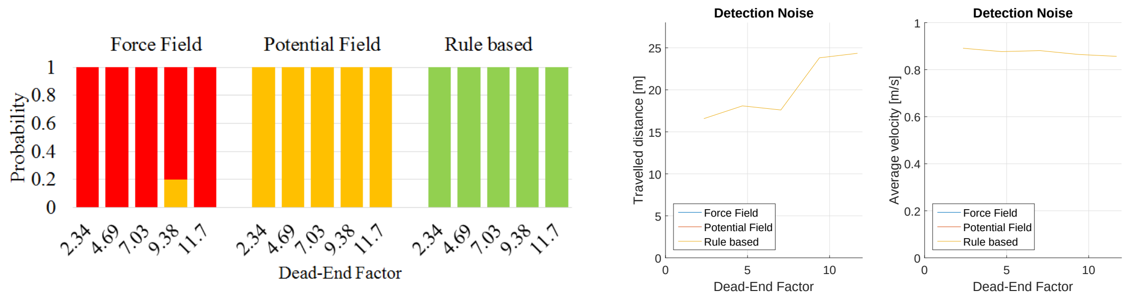


Figure 2-19: Results dead-end percentage simulations with detection noise

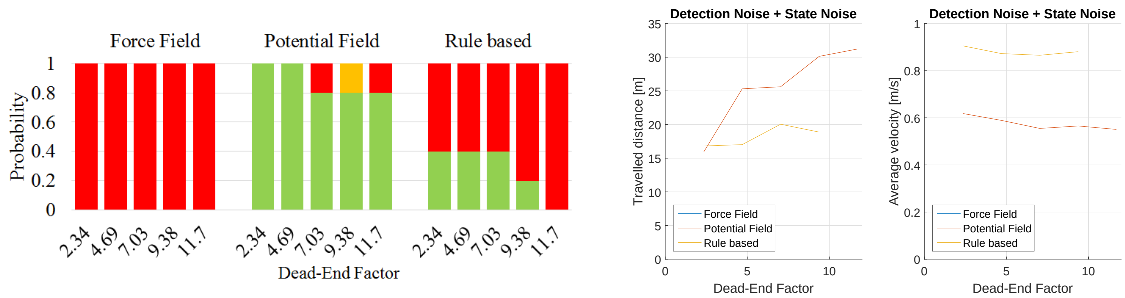


Figure 2-20: Results dead-end percentage simulations with detection and state noise

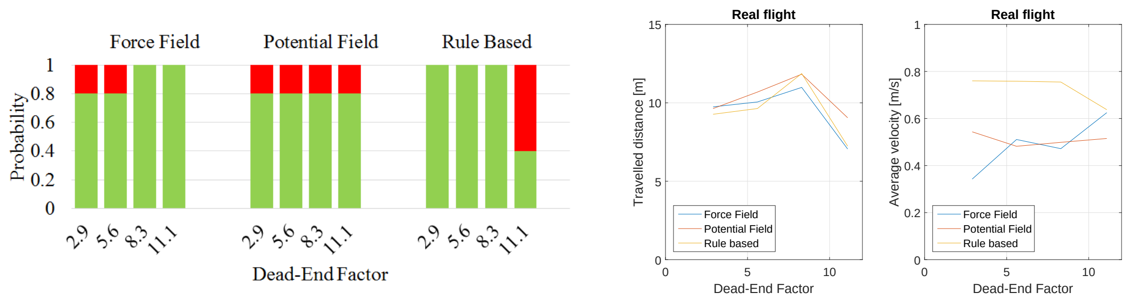


Figure 2-21: Results dead-end percentage real-flight tests

Results of the 'Average Orientation Angle' tests

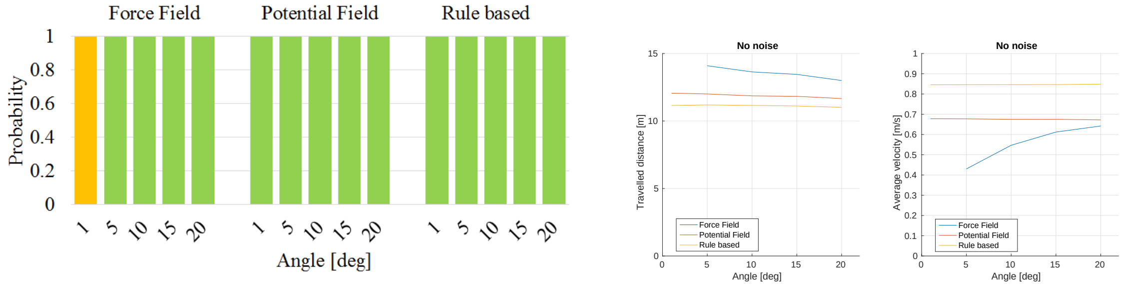


Figure 2-22: Results average orientation angle simulations without noise

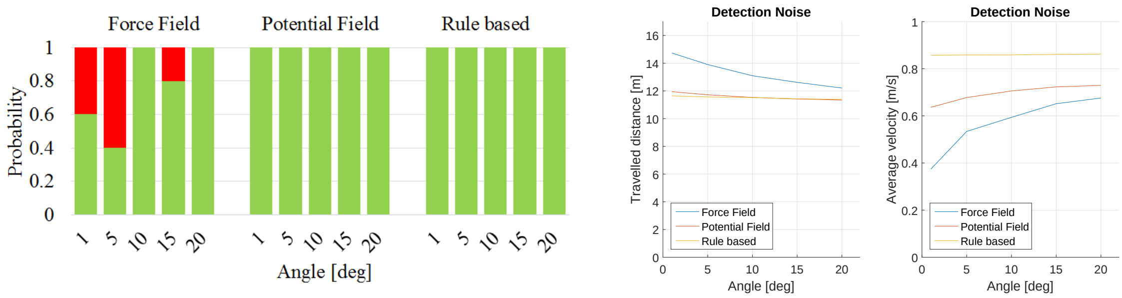


Figure 2-23: Results average orientation angle simulations with detection noise

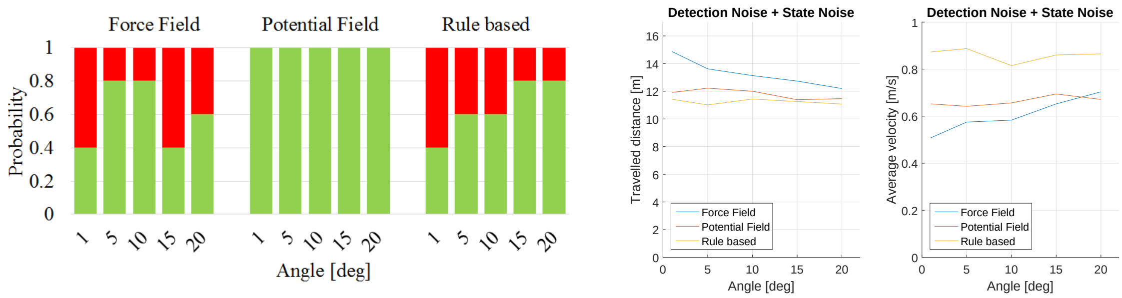


Figure 2-24: Results average orientation angle simulations with detection and state noise

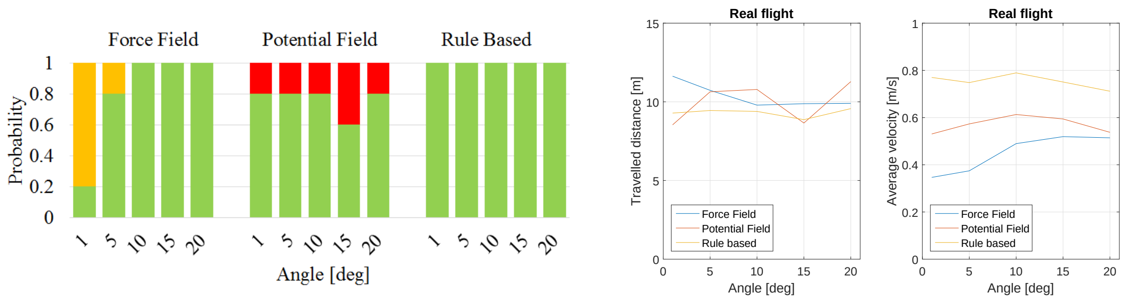


Figure 2-25: Results average orientation angle real-flight tests

3 Bootstrap hypothesis test

Confidence intervals

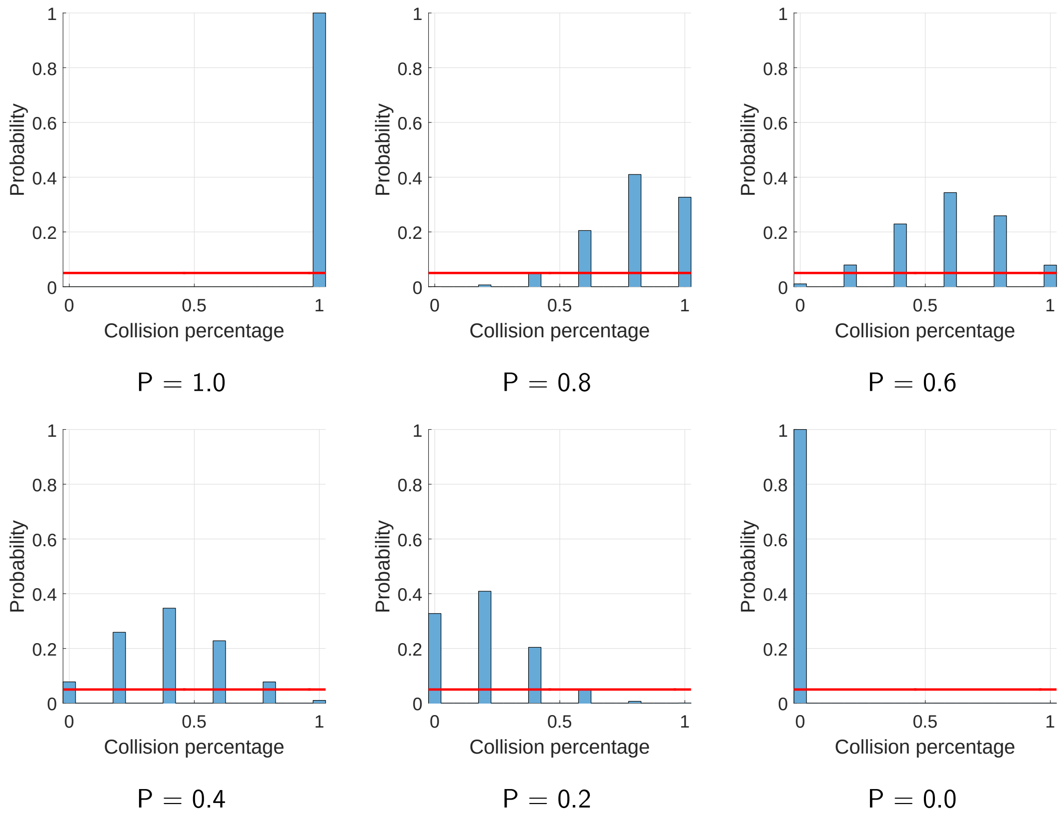


Figure 3-26: Bootstrap analysis

Hypothesis test

TABLE: Hypothesis test between two avoidance methods with a certain collision percentage.
✗No significant difference, ✓significant difference.

	0	20	40	60	80	100
0	✗	✗	✓	✓	✓	✓
20	✗	✗	✗	✓	✓	✓
40	✓	✗	✗	✗	✓	✓
60	✓	✓	✗	✗	✗	✓
80	✓	✓	✓	✗	✗	✗
100	✓	✓	✓	✓	✗	✗

Part V

Overview tables

Table 1-1: Overview of OA methods

	Basic			Environment					Detection			Conflict detection						Avoidance			
	Single(1)/Multiple(2)	combined(1) / seperate OA(2)	(online) mapping(1)/ no mapping(2)	Unknown(1)/ Known(2)	Static Obst. Yes(1)/No(2)	Dynamic Obst. Yes(1)/No(2)	Intelligent Obst. No communication Yes(1)/ No(2)	Intelligent Obst. With communication Yes(1)/ No(2)	sensor type(s)	Method to calculate parameters	parameter measured	Explicit conflict detection Yes(1)/ No(2)	dynamic constraints Yes(1)/ No(2)	obstacle uncertainty yes(1)/no(2)	collision flag(1)/ collision set(2)	state dimension	state propagetion	dynamic constraints Yes(1)/ No(2)	multiple(1) /single(2) obst. Taken into account	avoidance method	Control actions(Velocity/turn/both)
Both																					
Fiorini1998	1	1	2	1	1	1	2	2			Pos, vel	1	2	2	2	2D	Worst Case	1	1	Hearistic Search	Velocity Vector
Shah2009	2	2	2	1	1	1	2	2	Vision		Pos	1	2	2	1	3D	Flight plan	2	2	Increased curvature in PH path planning	
Shah2010	2	2	2	2	1	2	2	2				2				3D		1	1	PH path planning	
Magree2014	1	2	1	1	1	2	2	2	Monocular vision		height	1	2	2	1	3D	Staight	2	1	Predfined rules	Altitude
Boivin2008	2	1	1	1	1	2	2	1			Centrel pos, Shape	1	1	2	1	3D	Flight plan	1	1	Predictive control	-
Chen2013	1		1	1	1	1	2	2				1	1	1	1	2D	Flight plan	1	1	Potential field	Velocity, Heading
Sch2005	1	2	2	1	1	2	2	2	Stereo Vision		distance	2				3D		2	2	Behaviour based	Velocity, Heading
Ross2013	1	2	2	1	1	2	2	2	Monocular vision			2				3D		2	1	Dagger/reinforced learning	Heading
Marlow2011	1	1	1	1	1	2	2	2	Monocular vision		range/covariance	2	2	1		2D		2	1	Potential field	Heading
Sensors																					
wu2014									Monocular vision												
Kwag2007, Kwag2004, Kwag2007a									radar sensor												
Borenstein1988	1	2	1	1	1	2	2	2	Ultrasonic range finders		distance	2				2D		2	1	Path planning	Heading
Detection																					
Frew2004	1	2	2	1	1	2	2	2	Stereo Vision		Pos	1	1	1	2	2D	Worst Case		1		
Carlson[19],Albaker2010														1							
Merz[20], Albaker2010														1							
Ferrick2012	1	2	2	1	1	2	2	2	Lidar		distance	1	2	1	1	2D	Flight plan				
Karaman2012	1	1	2	2	1	2	2	2				2				2D		1	1	No method, assumed optimal can be calculated	
Ficuciello2013	1		2	2	2	1	2	2	Lidar/Webcam		distacne/goal	1	2	1		2D		2	1	MDP	Power
Gosiewki2011	1			1	1	2	2	2	Monocular vision	Optical flow, Lucas-Kanade gradient method	depth					3D					
Saha2014	1	2	2	1	1	2	2	2	Monocular vision			2				3D					
Prevost2007	2			1	1	1	2	2		EKF	position, speed, trajectory								2		
Resolution																					
Ariyur2005	1	2	2	1	1	2	2	2	radar		distance	2				3D		1	2	vector method	Velocity
Delin2012	1	2	2	1	1	2	2	2	SLRF		distance	2				2D		2	1	Predefined rules	Velocity, Heading
Huang2006	1	1	2	1	1	2	2	2	Single Camera Vision		Angular with of obstacles, Relative heading	2				2D		2	1	Potential field	Yaw rate
Geng2013	1	2	2	1	1	2	2	2			distance,										
Alexopoulos2013, Park[3]	1	2	2	1	1	1	1	1				1	2	2	2	2D	linear	2	2	Geometric approach	Heading
Alexopoulos2013,ref[4,5,13]	1	2	2	1	1	2	2	2				2				2D		2	1	repulsive force field	
Alexopoulos2013,ref[6,7,14]	1	2	2	1	1	2	2	2				2				2D		2	1	Online path re-planning	
Alexopoulos2013, Rathbun[8]												2				2D				Evolutionary approach	
Dong2005	1	2	2	1	1	1	2	2			distance, angle	2				3D		2	1	Fuzzy logic	Velocity vector
Alejo2014	2	1	2	2	1	1	1	1			relative pos, vol	1	2	2	2	3D	Worst Case	1	1	Geometric(RCA)	Velocity, Heading
Alejo2014, Lavalle[6]																				RRT	
Alejo2014, ref[7,8,9,11]																				evolutionary techniques	
Alejo2014, Pontani[10]																				particle swarm optimization	
Alejo2014, Stentz[12]																				graph search(A*, D*)	
Alejo2014, Lalish[17]																				Potential field	
Dario2014	1	2	2	1	1	2	2	2	ultrasound		distance	2				3D		2	1	Fuzzy logic	Velocity vector
Michels2005	1	2	2	1	1	2	2	2	Monocular vision	Reinforced learning	Depth(distance)	2				3D		2	2	Reinforced learning	Velocity, Heading
Michels2005, Scharstein & Szeliski2002									Stereo Vision	NE											
Schmitt2014	1	2	2	1	1	1	2	2				1	2	2	2	3D	linear	2	1	Geometric	
Schmitt2014,ref[1-3]					1																
Schmitt2014,ref[4-7]						1															
Schmitt2014,ref[8,9]															2						
Schmitt2014,ref[2,5]																				Aimpoint-based	
Hrabar2011	1	2	1	1	1	2	2	2	Stereo/ Laser		Range	1	2	1	2	3D	linear			Elliptical search to find Escape point	
Hrabar2008	1	1	1	1	1	2	2	2	Stereo vision		Position(Map)	2				3D		2	1	Probability road maps(graph) with D*	
Hrabar2008,ref[4-6]									Stereo vision	Optic flow											
Hrabar2008,ref[9,10]		1							Laser												
Hrabar2008, ref[11,12,13]									Stereo Vision												
Joelianto2013	1	1	2	1	1	2	2	2				2				3D		1	2	Model predictive control	
Jongho2012	1	2	2	1	1	2	2	2	Stereo Vision	Disparity	Depth	2				2D		2	2	Geometric/aiming point	
Kandil2010	1	2	2	1	1	1	2	2	radar		Distance/ Relative speed	2				2D		2	1	Potential field	Velocity
Kandil2010, ref[18, 20, 21]																				Potential field	
Lee2011	1	1	2	1	1	1	2	2	Monocular Vision		Range	2				3D		1	1	Model predictive control	
Muller2014	1	2	1	1	1	2	2	2	Ultra sonic		Distance	2				2D		2	1	Predfined rules	
Muller2014, ref[2,3]									Laser scanner												
Muller2014, ref[4,5,7]									Vision												
Muller2014, ref[6]									Combination!												
Sebesta2012	1	2	2	1	1	2	2	2	Monocular Vision	Optical flow/ Lucas Kanada	transit time	2				2D		1	1	Predefined rules	Heading
Stowers2011	1	2	2	1	1	2	2	2	Monocular vision	Optical flow/ Lucas Kanada	Depth	2				3D		2	1	Force field	Heading, Pitch
Watanabe2007	1	2	2	1	1	2	2	2	Monocular vision	EKF	position	1	2	2	2	2D		1	1	Aimingpoint/MEG(minimum-effert guidance)	
Yang2014a	1	1	2	2	1	2	2	2				2				2D		2	1	Genetic algorithm	
Tilove1990	1																				
Infinium20..		2	2	1	1	1	1	2	Stereo vision		distance	?	?	?		?	?	?	?	?	?

Table 1-2: Overview of evaluation methods

	Detection performance			*Resolution Performance*									
	Detection performance	Environment type	Performance metric	Real(1)/ Simulated(2)/ Both(B)	Indoors(1)/ Outdoors(2)	dense(1)/ sparse(2)	Static Obst. Yes(1)/ No(2)	Dynamic Obst. Yes(1)/ No(2)	Intelligent Obst./ No communication Yes(1)/ No(2)	Intelligent obst. With communication Yes(1)/ No(2)	# test	Testing	Performance parameter
Both													
Fiorini1998							1	1	2	2			Successful avoidance
Shah2009							1	2	2	2			
Shah2010	2			2	1	2	2	2	1	2			Curvature
Magree2014							1	2	2	2			Successful avoidance
Boivin2008							1	2	2	2			Successful avoidance+Computational Cost
Chen2013							1	2	2	2			Successful avoidacne/ optimal path
Sch2005					2		1	2	2	2			
Ross2013					2		1	2	2	2			successful avoidance
Marlow2011							1	2	2	2			successful avoidance
Sensors													
wu2014													
Kwag2007, Kwag2004, Kwag2007a													
Borenstein1988	2				1	2	1	2	2	2			successful avoidance
Detection													
Frew2004													
Carlson[19],Albaker2010													
Merz[20], Albaker2010													
Ferrick2012					1		1	2	2	2			Successful avoidance/ Processing time
Karaman2012	2				2	1	1	2	2	2	200 per V/rho pair	Trees with increased density and speed	Existance of path
Ficuciello2013	1				2	2	2	1	2	2	4500	Pedestrians random starting to random goal	Successful avoidance\
Gosiewki2011													
Saha2014	1	Inside one obstacle	Error + variance, succesfull detection										
Prevost2007	1		Variance of prediction error, error										
Resolution	1												
Ariyur2005							1	2	2	2			Successful avoidance
Delin2012					1		1	2	2	2			Successful avoidance
Huang2006							1	2	2	2			Successful avoidance
Geng2013							1	2	2	2			Path optimality
Alexopoulos2013, Park[3]							1	1	2	2			Successful avoidance
Alexopoulos2013,ref[4,5,13]							1	1	2	2			Successful avoidance
Alexopoulos2013,ref[6,7,14]							1	1	2	2			Successful avoidance
Alexopoulos2013, Rathbun[8]													
Dong2005							1	2	2	2			Successful avoidance
Alejo2014							1	2	2	1			Successful avoidance+Computational Cost
Alejo2014, Lavalle[6]													
Alejo2014, ref[7,8,9,11]													
Alejo2014, Pontani[10]													
Alejo2014, Stentz[12]													
Alejo2014, Lalish[17]													
Dario2014							1	2	2	2			
Michels2005	1				2		1	2	2	2		Four environments	Time to collision/Average speed
Michels2005, Scharstein & Szeliski2002													
Schmitt2014	2			2	1	2	1	1	2	2		Three cases	
Schmitt2014,ref[1-3]													
Schmitt2014,ref[4-7]													
Schmitt2014,ref[8,9]													
Schmitt2014,ref[2,5]	1	Outside flight	Computation time										
Hrabar2011	2			B	2	2	1	2	2	2			Compuational time/ Succesfull avoidance
Hrabar2008				B	1	2	1	2	2	2	27(sim)/16(real)	5 obstacles in room(sim)/4 obstacles(real)	Succesful avoidance
Hrabar2008,ref[4-6]													
Hrabar2008,ref[9,10]													
Hrabar2008, ref[11,12,13]													
Joelianto2013	2			2	1	2	1	2	2	2		3 obstacle	
Jongho2012	2			2	1	2	1	2	2	2	one	1 obstacle	path/ distance to obstacle
Kandil2010	1	outside/Multiple objects	TP/FP										
Kandil2010, ref[18, 20, 21]				B									Accuracy/ Robustness
Lee2011	2			2	1	2	1	1	2	2		2 obstacles	Compared to PID / optimal trajectory
Muller2014	1		Range, Mean error, Standard deviation	1	1	2	1	2	2	2		Walls	Position error
Muller2014, ref[2,3]													
Muller2014, ref[4,5,7]													
Muller2014, ref[6]													
Sebesta2012	2			B	-	1	1	2	2	2		Markovian obstacle field	
Stowers2011	2			B	1	2	1	2	2	2		Use a dataset to which the alogrithm is applied and a flight test with a similar environment as the data-set(Corridor with people)	Succesfull avoidance
Watanabe2007	1	two obstalces	z-test, estimation error, std	2	1	2	1	2	2	2	2	Two obstacles	Distance from obstacle
Yang2014a	2			2	2	2	1	2	2	2	5	50km square, 3 no fly zones	Compuational effort
Tilove1990													

Table 1-3: Overview table Kuchar et al.

Model	Dimensions	Detection	Resolution	Maneuvers	Multiple
Andrews [32]	H	—	O	T	P
Bilimoria [33]	H	—	O	C(ST)	P & G
Chakravarthy [19]	H	—	O	C(ST)	P
Frazzoli [34]	H	—	O	C(ST)	G
Tomlin [35]	H	—	O	T	G
Irvine [36]	HV	—	O	C(STV)	P
Ota [37]	HV	—	O	C(TV)	G
Kosecka [38]	H	—	F	C(ST)	G
Zeghal [4]	H	—	F	C(ST)	G
Eby [39,40]	HV	—	F	C(STV)	G
Sridhar [41]	H	√	—	—	P
EGPWS [12]	HV	√	—	—	—
Havel [42]	HV	√	—	—	P
Kelly [3]	HV	√	—	—	P
TCAD [15]	HV	√	—	—	P
GPWS [11]	V	√	P	V	—
PRM [13]	H	√	P	C(TV)	P
Bilimoria [43]	HV	√	P	STV	P
Burgess [44]	H	√	O	TV	P
Coenen [16]	H	√	O	ST	P
Gazit [45]	H	√	O	VT	P
Harper [46]	H	√	O	C(ST)	G
Iijima [17]	H	√	O	ST	P
Niedringhaus [47]	H	√	O	C(ST)	G
Zhao [48]	H	√	O	T	P
Burdun [49]	HV	√	O	C(STV)	P
Durand [50]	HV	√	O	T	G
Ford [51]	HV	√	O	V	P
Krozel [6,52]	HV	√	O	STV	P
Love [53]	HV	√	O	TV	P
Menon [54]	HV	√	O	C(STV)	G
Niedringhaus [55]	HV	√	O	STV	G
Schild [56]	HV	√	O	C(TV)	P
TCAS [14]	HV	√	O	V	P
Hoekstra [24]	HV	√	F	C(STV)	P
Zeghal [57]	HV	√	F	C(STV)	G
Duong [23]	HV	√	M & F	C(STV)	P

Table 1-4: Overview table Bonin-Font et al.

Summary of the most outstanding visual navigation studies from the late 90's to the present

Authors	Type of vehicle	Category	Strategy	Type of visual sensor
[121, 122]	Ground	Map building	Visual SLAM. Landmarks localization and tracking	Single standard camera
[123, 124]	Ground	Map building	Visual SLAM. Landmarks extraction and occupancy grids	Stereo cameras
[25]	Ground	Map building	Visual SLAM. Map feature extraction	Single standard camera
[26]	Ground	Map building	Visual SLAM. 3D sparse mapping of interest points	Single wide angle camera
[81]	Ground	Map building	3D construction of an occupancy grid	Single standard camera
[139]	Ground	Map building	3D high density map and object recognition	Single standard camera
[67]	Ground	Map building	Human guided pre-training	Stereo cameras
[111]	Ground	Map building	Human guided pre-training	Single wide angle camera
[147]	Ground	Map building	Topological map	Omnidirectional camera
[42]	Ground	Map building	Topological map	Omnidirectional camera
[70, 108]	Ground	Map building	Topological map	Single standard camera
[6]	Ground	Map building	Local occupancy grid	Stereo cameras
[39]	Ground	Map building	Local occupancy grid	Single standard camera
[45]	Ground	Map building	Local occupancy grid	Stereo cameras
[40]	Ground	Map building	Local occupancy grid	Single standard camera
[17,32,73,84,85]	Ground	Map building	Visual sonar	Single standard camera
[113]	Ground	Mapless	Optical flow	Single standard camera
[14]	Ground	Mapless	Optical flow	Single wide angle camera
[131, 132]	Ground	Mapless	Optical flow combined with stereo information	Stereo cameras
[133]	Ground	Mapless	Optical flow	Single standard camera
[87, 89, 107]	Ground	Mapless	Appearance-based method	Standard or omnidirectional single camera
[98]	Ground	Mapless	Appearance-based method	Panoramic camera
[138]	Ground	Map building	Museum guiding robot: complete 3D map	Single standard camera
[118]	Ground	Map building	Museum guiding robot: topological map	Single standard camera
[15, 76]	Ground	Mapless	Image qualitative characteristics extraction	Single standard camera
[80]	Ground	Mapless	Image qualitative characteristics extraction	Single standard camera
[61]	Ground	Mapless	Image qualitative characteristics extraction	Stereo cameras
[24, 75, 105, 151]	Ground	Mapless	Features tracking; homography	Single standard camera
[112]	Ground	Mapless	Features tracking; homography	Stereo cameras
[78, 116]	Ground	Mapless	Features tracking: SIFT	Single standard camera
[100, 145]	UAV	Mapless	Optical flow: insect inspired (EMD)	Camera eye
[127]	UAV	Mapless	Optical flow: insect inspired (EMD)	Single standard camera
[52, 53]	UAV	Mapless	Optical flow: insect inspired (EMD)	Single mini wireless camera
[62]	UAV	Mapless	Optical flow: insect inspired	Stereo cameras looking forward combined with two sideways looking cameras
[93]	UAV	Mapless	Features tracking	Single wide angle camera
[30, 43]	Amphibious	Map building	Visual SLAM. 3D total map building	Trinocular stereo cameras
[102]	AUV	Mapless	Features tracking; ho-	Single standard camera

Table 1-5: Table Goerzen et al. without constraints

Table 1 Comparison of algorithms for path planning without differential constraints

Algorithm name	Type of problem solved	Number of dimensions	Completeness	Optimality	Proven time complexity
Roadmap					
Visibility graph	Point vehicle	2	Complete	Pptimal	$O(N^2)$
Edge-sampled visibility graph	Point vehicle	3	Complete	Resolution optimal	Exact solution is NP hard, approximation polynomial in N and 1/eta
Voronoi roadmap	Point vehicle	2	Complete	Non-optimal	$O(N \log N)$
Freeway method	Point vehicle	2	Incomplete	Non-optimal	
Silhouette method	Point vehicle	Arbitrary	Complete	Non-optimal	$O(\mathrm{Exp}[D,N])$
Exact cell decomposition					
Trapezoidal decomposition	Point vehicle	2	Complete	Non-optimal	$O(N \log N)$
Critical curves and non-critical regions	Regid vehicle	2	Complete	Non-optimal	$O(N^2 \log N)$
Cylindrical algebraic decomposition	Regid vehicle	Arbitrary	Complete	Bon-optimal	$O(\mathrm{Exp}(\mathrm{Exp}(N)), \mathrm{poly}(P)$
Approximate cell decomposition					
Rectanguloid	Point vehicle	2	Resolution complete	Non-optimal	
Quadtreee, octree, or 2^m tree decomposition	Point vehicle	Arbitrary	Resolution complete	Non-optimal	
Approximate and decompose	Point vehicle	2	Resolution complete	Non-optimal	
Potential-based methods					
Potential field	All types	Arbitrary	Incomplete	Non-optimal	
Potential guided	All types	Arbitrary	Complete	Non-optimal	
Harmonic potential functions	Point vehicle	Arbitrary	Complete	Non-optimal	$O(M \log M)$
Wavefront expansion	Regid vehicle	Arbitrary	Resolution complete	Resolution optimal	$O(M \log M)$
Wavefront expansion with a skeleton (NF2)	Regid vehicle	Arbitrary	Resolution complete	Non-optimal	$O(M \log M)$
Continuous Dijkstra	Regid vehicle	2	Complete	Optimal	$O(N^2 \log N)$
Probabilistic approaches					
Randomized potential fields	All types	Arbitrary		Non-optimal	
Global learning	All types	Arbitrary	Probabilistically complete when complete search used	Non-optimal	
Weighted region problem					
Exact algorithm	Polygonal weighted regions	2	Complete	Optimal	$O(N^8 \log N)$
Approximation algorithm	Polygonal weighted regions	2	Complete	Resolution optimal	$O(N^* M \log N^* M)$
Geodesic distance on a grid	Weighted grid	Arbitrary	Resolution complete	Resolution optimal	$O(M \log M)$

Table 1-6: Table Goerzen et al. with constraints

Table 2 Comparison of algorithms for trajectory planning with differential constraints

Algorithm name	Completeness	Optimality	Soundness	Proven time complexity
Canonical two-dimensional solution	Complete	Optimal	Sound	$O(\exp(N))$
State-space sampling				
State space lattice search	Resolution complete	Resolution optimal	Resolution sound	$O(C^d N\mathrm{err}^{(-6d)})$
Dynamic programming with interpolation	Resolution complete	Resolution optimal	Resolution sound	$O(C^d N\mathrm{err}^{(-3d)})$
Rapidly-expanding random tree (RRT)	Probabilistically complete	Non-optimal	Resolution sound	
Reachability graph, rapidly-expanding dense tree (RDT)	Resolution complete	Resolution optimal	Resolution sound	
Pruned reachability graph	Resolution complete	Resolution optimal	Resolution sound	
Minimum distance path followed by trajectory forming				
Linked boundary point solvers	Resolution complete			
Maneuver automation	Resolution complete when complete search used		Resolution sound if constraints are met	
Decoupled approach		Non-optimal		$O(\mathrm{polynomial}(N))$
Plan and transform (spline smoothing)		Non-optimal		
Path-constrained trajectory planning	Resolution complete when complete search used	Non-optimal	Resolution sound	
Plane slicing	Resolution complete	Non-optimal	Resolution sound	
Mathematical programming				
Gradient-based trajectory optimization	Resolution complete when complete search used	Resolution optimal when complete search used	Resolution sound (for infinite horizon)	$O(\exp(M))$
Model predictive control (receding horizon)			Resolution sound if constraints are met	
Potential-based				
Path-constrained trajectory planning	Resolution complete when complete navigation function is used	Non-optimal	Resolution sound if space varying speed limit constraints are met	$O(M \log M)$
Randomized potential fields		Non-optimal		

Table 1-7: Comparison using three performance measures

	Tsukuba			Sawtooth			Venus			Map	
	$B_{\overline{\mathcal{O}}}$	$B_{\overline{\mathcal{T}}}$	$B_{\mathcal{D}}$	$B_{\overline{\mathcal{O}}}$	$B_{\overline{\mathcal{T}}}$	$B_{\mathcal{D}}$	$B_{\overline{\mathcal{O}}}$	$B_{\overline{\mathcal{T}}}$	$B_{\mathcal{D}}$	$B_{\overline{\mathcal{O}}}$	$B_{\mathcal{D}}$
20 Layered	1.58 *3*	1.06 *4*	8.82 *3*	**0.34** *1*	**0.00** *1*	**3.35** *1*	1.52 *3*	2.96 *10*	2.62 *2*	0.37 *6*	5.24 *6*
*4 Graph cuts	1.94 *5*	1.09 *5*	9.49 *5*	1.30 *6*	0.06 *3*	6.34 *6*	1.79 *7*	2.61 *8*	6.91 *4*	0.31 *4*	3.88 *4*
19 Belief prop.	**1.15** *1*	**0.42** *1*	**6.31** *1*	0.98 *5*	0.30 *5*	4.83 *5*	1.00 *2*	0.76 *2*	9.13 *6*	0.84 *10*	5.27 *7*
11 GC+occl.	1.27 *2*	0.43 *2*	6.90 *2*	0.36 *2*	**0.00** *1*	3.65 *2*	2.79 *12*	5.39 *13*	**2.54** *1*	1.79 *13*	10.08 *12*
10 Graph cuts	1.86 *4*	1.00 *3*	9.35 *4*	0.42 *3*	0.14 *4*	3.76 *3*	1.69 *6*	2.30 *6*	5.40 *3*	2.39 *16*	9.35 *10*
8 Multiw. cut	8.08 *17*	6.53 *14*	25.33 *18*	0.61 *4*	0.46 *8*	4.60 *4*	**0.53** *1*	**0.31** *1*	8.06 *5*	0.26 *3*	3.27 *3*
12 Compact win.	3.36 *8*	3.54 *8*	12.91 *9*	1.61 *9*	0.45 *7*	7.87 *7*	1.67 *5*	2.18 *4*	13.24 *9*	0.33 *5*	3.94 *5*
14 Realtime	4.25 *12*	4.47 *12*	15.05 *13*	1.32 *7*	0.35 *6*	9.21 *8*	1.53 *4*	1.80 *3*	12.33 *7*	0.81 *9*	11.35 *15*
*5 Bay. diff.	6.49 *16*	11.62 *19*	12.29 *7*	1.45 *8*	0.72 *9*	9.29 *9*	4.00 *14*	7.21 *16*	18.39 *13*	**0.20** *1*	2.49 *2*
9 Cooperative	3.49 *9*	3.65 *9*	14.77 *11*	2.03 *10*	2.29 *14*	13.41 *13*	2.57 *11*	3.52 *11*	26.38 *17*	0.22 *2*	**2.37** *1*
*1 SSD+MF	5.23 *15*	3.80 *10*	24.66 *17*	2.21 *11*	0.72 *10*	13.97 *15*	3.74 *13*	6.82 *15*	12.94 *8*	0.66 *8*	9.35 *10*
15 Stoch. diff.	3.95 *10*	4.08 *11*	15.49 *15*	2.45 *14*	0.90 *11*	10.58 *10*	2.45 *9*	2.41 *7*	21.84 *15*	1.31 *12*	7.79 *9*
13 Genetic	2.96 *6*	2.66 *7*	14.97 *12*	2.21 *12*	2.76 *16*	13.96 *14*	2.49 *10*	2.89 *9*	23.04 *16*	1.04 *11*	10.91 *14*
7 Pix-to-pix	5.12 *14*	7.06 *17*	14.62 *10*	2.31 *13*	1.79 *12*	14.93 *17*	6.30 *17*	11.37 *18*	14.57 *10*	0.50 *7*	6.83 *8*
6 Max flow	2.98 *7*	2.00 *6*	15.10 *14*	3.47 *15*	3.00 *17*	14.19 *16*	2.16 *8*	2.24 *5*	21.73 *14*	3.13 *17*	15.98 *18*
*3 Scanl. opt.	5.08 *13*	6.78 *15*	11.94 *6*	4.06 *16*	2.64 *15*	11.90 *11*	9.44 *19*	14.59 *19*	18.20 *12*	1.84 *14*	10.22 *13*
*2 Dyn. prog.	4.12 *11*	4.63 *13*	12.34 *8*	4.84 *19*	3.71 *19*	13.26 *12*	10.10 *20*	15.01 *20*	17.12 *11*	3.33 *18*	14.04 *17*
17 Shao	9.67 *18*	7.04 *16*	35.63 *19*	4.25 *17*	3.19 *18*	30.14 *20*	6.01 *16*	6.70 *14*	43.91 *20*	2.36 *15*	33.01 *20*
16 Fast Correl.	9.76 *19*	13.85 *20*	24.39 *16*	4.76 *18*	1.87 *13*	22.49 *18*	6.48 *18*	10.36 *17*	31.29 *18*	8.42 *20*	12.68 *16*
18 Max surf.	11.10 *20*	10.70 *18*	41.99 *20*	5.51 *20*	5.56 *20*	27.39 *19*	4.36 *15*	4.78 *12*	41.13 *19*	4.17 *19*	27.88 *19*

Bibliography

Albaker, B. M., & Rahim, N. a. (2010). Unmanned aircraft collision detection and resolution: Concept and survey. *Proceedings of the 2010 5th IEEE Conference on Industrial Electronics and Applications*, 248–253.

Alejo, D., Cobano, J. a., Heredia, G., & Ollero, a. (2014). Optimal Reciprocal Collision Avoidance with mobile and static obstacles for multi-UAV systems. *2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014 - Conference Proceedings*, 1259–1266.

Alexopoulos, A., Kandil, A., Orzechowski, P., & Badreddin, E. (2013). A Comparative Study of Collision Avoidance Techniques for Unmanned Aerial Vehicles. *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 1969–1974.

Amann, M.-C., Bosch, T., Lescure, M., Myllyla, R., & Rioux, M. (2001). Laser ranging: a critical review of usual techniques for distance measurement. *Optical Engineering*, *40*(1), 10.

Ariyur, K., Lommel, P., & Enns, D. (2005). Reactive inflight obstacle avoidance via radar feedback. *Proceedings of the 2005, American Control Conference, 2005.*, 2978–2982.

Bachrach, A., He, R., & Roy, N. (2010). Autonomous Flight in Unknown Indoor Environments. *International Journal of Micro Air Vehicles*, *1*(4), 217–228.

Barron, J. L., Fleet, D. J., & Beauchemin, S. S. (1994). Performance of Optical Flow Techniques. *International journal of computer science*, *1*(12), 43–77.

Benet, G., Blanes, F., Simó, J. E., & Pérez, P. (2002). Using infrared sensors for distance measurement in mobile robots. *Robotics and Autonomous Systems*, *40*(4), 255–266.

Boivin, E., Desbiens, A., & Gagnon, E. (2008). UAV collision avoidance using cooperative predictive control. *16th Mediterranean Conference on Control and Automation*, 682–688.

Bonin-Font, F., Ortiz, A., & Oliver, G. (2008). Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*.

Borenstein, J., & Koren, Y. (1988). Obstacle avoidance with ultrasonic sensors. *IEEE Journal on Robotics and Automation*, *4* (2), 213–218.

Borenstein, J., & Koren, Y. (1991). The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots. *IEEE Journal of Robotics and Automation*, *7* (3), 278–288.

Bowyer, K., & Phillips, P. (1998). Overview of work in empirical evaluation of computer vision algorithms. , 1–11.

Brady, J. (2006). Automating the testing process of image processing.

Chen, H., Chang, K., & Agate, C. S. (2013). UAV path planning with tangent-plus-lyapunov vector field guidance and obstacle avoidance. *IEEE Transactions on Aerospace and Electronic Systems*, *49* (2), 840–856.

Croon, G.C.H.E. de, C. D. W. (2011). Sky Segmentation Approach to Obstacle Avoidance. *Ieeeac*, 4244–7351.

Darío, R., Tarazona, F., & Lopera, F. R. (2014). Anti-collision System for Navigation Inside an UAV Using Fuzzy Controllers and Range Sensors.

Davies, E. (2012). *Computer & Machine vision* (Fourth ed.). Oxford: Elsevier B.V.

Delin, L., Fei, W., Biao, W., & Chen, B. M. (2012). Implementation of obstacle avoidance technique for indoor coaxial rotorcraft with Scanning Laser Range Finder. *Control Conference (CCC), 2012 31st Chinese*, 5135–5140.

Desouza, G. N., & Kak, A. C. (2002). Vision for Mobile Robot Navigation : A Survey. , *24* (2), 237–267.

Dong, T. D. T., Liao, X., Zhang, R., Sun, Z. S. Z., & Song, Y. (2005). Path Tracking and Obstacles Avoidance of UAVs - Fuzzy Logic Approach. *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, 43–48.

Fasano, G., Accardo, D., Moccia, a., & Paparone, L. (2006). Airborne multisensor tracking for autonomous collision avoidance. *2006 9th International Conference on Information Fusion, FUSION*.

Ferrick, A., Fish, J., Venator, E., & Lee, G. S. (2012). UAV obstacle avoidance using image processing techniques. *2012 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, 73–78.

Ficuciello, F., Palli, G., Melchiorri, C., & Siciliano, B. (2013). Experimental Robotics. , *88*, 515–529.

Fiorini, P., & Shiller, Z. (1998). Motion Planning in Dynamic Environments using Velocity Obstacles. *I. J. Robotic Res*, *17*, 760–772.

Frew, E., & Sengupta, R. (2004). Obstacle avoidance with sensor uncertainty for small unmanned aircraft. *Proceedings of the IEEE Conference on Decision and Control*, *1*, 614–619.

Geng, Q., Shuai, H., & Hu, Q. (2013). Obstacle avoidance approaches for quadrotor UAV based on backstepping technique. *2013 25th Chinese Control and Decision Conference, CCDC 2013*, 3613–3617.

Goerzen, C., Kong, Z., & Mettler, B. (2010). *A survey of motion planning algorithms from the perspective of autonomous UAV guidance* (Vol. 57).

Gool, L. V., Szekely, G., & Ferrari, V. (2011). *Computer Vision.* London: Springer-Verlag.

Gosiewski, Z., Ciesluk, J., & Ambroziak, L. (2011). Vision-based obstacle avoidance for unmanned aerial vehicles. *2011 4th International Congress on Image and Signal Processing*, *4*(4), 2020–2025.

Guzel, M., & Bicker, R. (2009). Vision Based Obstacle Avoidance Techniques. *Intechopen.Com*.

Holenstein a.a., & Badreddin, E. (1991). Collision avoidance in a behavior-based mobile robot design. *Proceedings. 1991 IEEE International Conference on Robotics and Automation*(April), 898–903.

Hrabar, S. (2008). 3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 807–814.

Hrabar, S. (2011). Reactive obstacle avoidance for rotorcraft UAVs. *IEEE International Conference on Intelligent Robots and Systems*, 4967–4974.

Huang, W. H., Fajen, B. R., Fink, J. R., & Warren, W. H. (2006). Visual navigation and obstacle avoidance using a steering potential function. *Robotics and Autonomous Systems*, *54*, 288–299.

Joelianto, E., Budiyono, A., Wijayanti, I. E., & Megawati, N. Y. (2013). Model Predictive Control for Obstacle Avoidance as. (Mld), 127–132.

Jongho, P., & Youdan, K. (2012). Stereo vision based collision avoidance of quadrotor UAV. *Control, Automation and Systems (ICCAS), 2012 12th International Conference on*, 173–178.

Kandil, A. a., Wagner, A., Gotta, A., & Badreddin, E. (2010). Collision avoidance in a recursive nested behaviour control structure for unmanned aerial vehicles. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 4276–4281.

Karaman, S., & Frazzoli, E. (2012). High-speed flight in an ergodic forest. *Proceedings - IEEE International Conference on Robotics and Automation*, 2899–2906.

Kemkemian, S., & Nouvel-Fiani, M. (2009). Radar systems for Sense and Avoid on UAV. *Radar Conference-Surveillance for a Safer World*, 1–6.

Koren, Y., & Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. *Proceedings. 1991 IEEE International Conference on Robotics and Automation*(April), 1398–1404.

Kuchar, J. K., & Kuchar, J. K. (2005). Safety Analysis Methodology for UAV Collision Avoidance Systems. *Regulation*.

Kuchar, J. K., & Yang, L. C. (2000). A review of conflict detection and resolution modeling methods. *IEEE Transactions on Intelligent Transportation Systems*, *1*(4), 179–189.

Kuttruff, H. (2009). *Room acoustics*.

Kwag, Y., & Kang, J. (2004). Obstacle awareness and collision avoidance radar sensor system for low-altitude flying smart UAV. *The 23rd Digital Avionics Systems Conference (IEEE Cat. No.04CH37576)*, *2*, 1–10.

Kwag, Y. K., Choi, M. S., Jung, C. H., & Hwang, K. Y. (2007). Collision avoidance radar for UAV. *CIE International Conference of Radar Proceedings*, 0–3.

Kytö, M., Nuutinen, M., & Oittinen, P. (2011). ¡title¿Method for measuring stereo camera depth accuracy based on stereoscopic vision¡/title¿. , 78640I–78640I–9.

Lazaros, N., Sirakoulis, G. C., & Gasteratos, A. (2008). Review of Stereo Vision Algorithms: From Software to Hardware. *International Journal of Optomechatronics*, *2*(4), 435–462.

Lee, D., Lim, H., & Kim, H. J. (2011). Obstacle avoidance using image-based visual servoing integrated with nonlinear model predictive control. *IEEE Conference on Decision and Control and European Control Conference*, 5689–5694.

Magree, D., Mooney, J. G., & Johnson, E. N. (2014). Monocular visual mapping for obstacle avoidance on UAVs. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *74*, 17–26.

Mahmood, F., Haider, S. M. B., & Kuwar, F. (2012). Investigating the Performance of Correspondence Algorithms in Vision based Driver-Assistance in Indoor Environment. *International Journal of Computer Applications*, *60*(9), 6–12.

Marlow, S. Q., & Langelaan, J. W. (2011). Local Terrain Mapping for Obstacle Avoidance Using Monocular Vision. *Journal of the American Helicopter Society*, *56*, 022007.

Matthies, L., Litwin, T., Owens, K., Rankin, A., Murphy, K., Coombs, D., et al. (1998). Performance evaluation of UGV obstacle detection with CCD/FLIR stereo vision and LADAR. *IEEE International Symposium on Intelligent Control (ISIC) held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell*, 658–670.

Meier, L., Tanskanen, P., Heng, L., Lee, G. H., Fraundorfer, F., & Pollefeys, M. (2012). PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, *33*(1-2), 21–39.

Mettler, B., Kong, Z., Goerzen, C., & Whalley, M. (2010). Benchmarking of obstacle field navigation algorithms for autonomous helicopters. *Proceedings of the 66th Annual Forum of the American Helicopter Society*, 1–18.

Moufid, H., Abielmona, R., Petriu, E., & Naji, K. (2008). Neural Control System of a Mobile Robot. *Neural Networks*, 2826–2833.

Mujumdar, A., & Padhi, R. (2011). Evolving Philosophies on Autonomous Obstacle/-Collision Avoidance of Unmanned Aerial Vehicles. *Journal of Aerospace Computing, Information, and Communication*, *8*(February), 17–41.

Müller, J., Ruiz, A. V., & Wieser, I. (2014). Safe & sound: A robust collision avoidance layer for aerial robots based on acoustic sensors. *Record - IEEE PLANS, Position Location and Navigation Symposium*, 1197–1202.

Park, M., Jeon, J., & Lee, M. (2001). Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing. *Industrial Electronics*, 1530–1535.

Petillot, Y., Ruiz, I. T., & Lane, D. M. (2001). Underwater vehicle obstacle avoidance and path planning using a muli-beam forward looking sonar. *IEEE Journal of Oceanic Engineering*, *26*(2), 240–251.

Reshetyuk, Y. (2006). Investigation and Calibration of Pulsed Time-of-Flight Terrestrial Laser Scanners. (October), p. 152.

Ross, S., Melik-Barkhudarov, N., Shankar, K. S., Wendel, A., Dey, D., Bagnell, J. A., et al. (2013). Learning monocular reactive UAV control in cluttered natural environments. *Proceedings - IEEE International Conference on Robotics and Automation*, 1765–1772.

Sabater, N., Morel, J.-M., & Almansa, a. (2011). How Accurate Can Block Matches Be in Stereo Vision? *SIAM Journal on Imaging Sciences*, *4*(1), 472–500.

Saha, S., Natraj, A., & Waharte, S. (2014). A Real-Time Monocular Vision-based Frontal Obstacle Avoidance for Low Cost UAVs. , 189–195.

Schafer, B. H., Proetzsch, M., & Berns, K. (2005). Stereo-Vision-Based Obstacle Avoidance in Rough Outdoor Terrain Challenges in Rough Terrain. *Most*, 1–9.

Scharstein, D., & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, *47*(1-3), 7–42.

Sebesta, K., & Baillieul, J. (2012). Animal-inspired agile flight using optical flow sensing. *Proceedings of the IEEE Conference on Decision and Control*(1), 3727–3734.

Shah, M. a., & Aouf, N. (2009). Dynamic cooperative perception and path planning for collision avoidance. *2009 6th International Symposium on Mechatronics and its Applications, ISMA 2009*, 1–7.

Stowers, J., Hayes, M., & Bainbridge-Smith, A. (2011). Biologically inspired UAV obstacle avoidance and control using monocular optical flow. *The 5th International Conference on Automation, Robotics and Applications*, 378–383.

Thacker, N. a., Clark, A. F., Barron, J. L., Ross Beveridge, J., Courtney, P., Crum, W. R., et al. (2008). Performance characterization in computer vision: A guide to best practices. *Computer Vision and Image Understanding*, *109*(3), 305–334.

Thiel, K., & Wehr, a. (2004). Performance Capabilities of Laser Scanners - an Overview and Measurement Principle Analysis. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XXXVI-87*(W2), 14–18.

Tilove, R. (1990). Local obstacle avoidance for mobile robots based on the method of artificial potentials. *Proceedings., IEEE International Conference on Robotics and Automation*.

Viswanathan, P., Boger, J., Hoey, J., & Mihailids, A. (2007). A Comparison of Stereovision and Infrared as Sensors for an Anti-Collision Powered Wheelchair for Older Adults with Cognitive Impairments. *2nd International Conference on Technology and Aging (ICTA)*.

Watanabe, Y., Calise, A. J., & Johnson, E. N. (2007). Vision-Based Obstacle Avoidance for UAVs. *Update*(August), 1–11.

Yang, W., & Wenjie, C. (2014). Path Planning and Obstacle Avoidance of Unmanned Aerial Vehicle Based on Improved Genetic Algorithms. , 8612–8616.

Young, I. T., Gerbrands, J. J., & Vliet, L. J. van. (1998). *Fundamentals of Image Processing (v.2.3)* (2.2 ed.). Delft: Delft University of Technology.