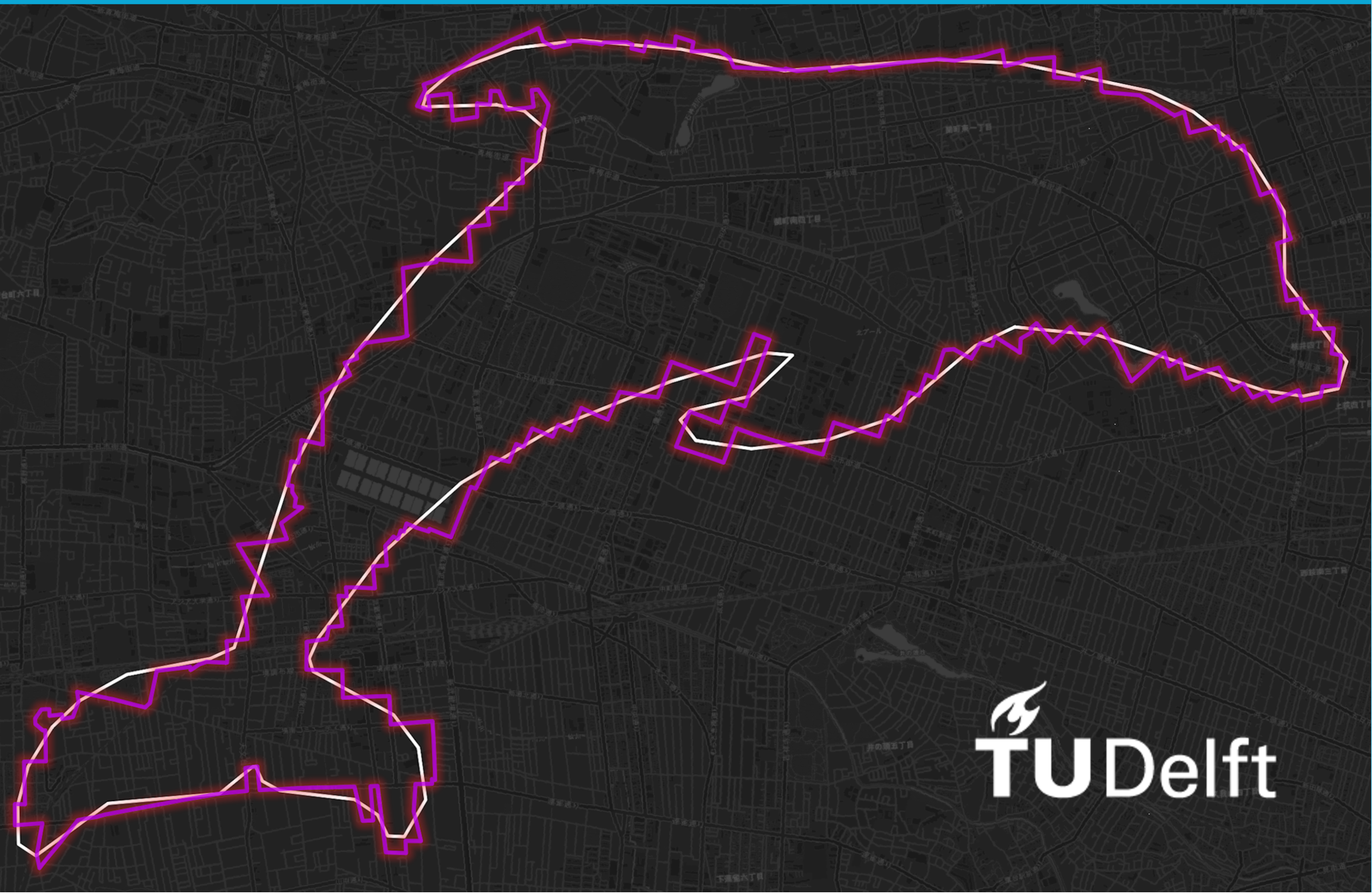
MSc thesis in Geomatics

# Shape-guided artistic route finding

Leon Powałka
2023



**MSc thesis in Geomatics**

# Shape-guided Artistic Route Finding

Leon Powałka

June 2023

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

Leon Powałka: *Shape-guided Artistic Route Finding* (2023)

The work in this thesis was carried out in the:

3D geoinformation group
Delft University of Technology

| | |
|---|---|
| Supervisors: | Dr. Liangliang Nan |
| | Dr. Jantien Stoter |
| Co-reader: | Dr. Michael Weinmann |

# Abstract

Creating GPS art on the map is an interesting way to make one's outdoor activity more engaging. Cyclists, runners and hikers can create impressive drawings on the map by traversing the road/pedestrian network in a carefully planned way. Such planning, however, is often tedious and time consuming, which makes the GPS artists have to meticulously design the routes with the complex road network in mind. The aim of our research is to come up with a full process that can express a person's initial idea (for example a contour drawing) as a route, which the user can then follow to create their GPS art. This involves transforming the input image to match the routing network in a selected area and generating a route which approximates the shape in the best possible way. In our work, searching for patterns in the road network is cast to an image matching problem with template matching as the solution. Generating routes is achieved using a graph routing algorithm with a custom cost function, to make the resulting route as similar to the input shape as possible. Finally, two ways of generating artistic routes are presented. First is an automatic GPS art workflow, which attempts to find an optimal initial location of the route, then generates a number of candidate routes and selects the best one according to various evaluation criteria. The second method is an interactive browser application, where the user can select an initial location for his shape on the map, move, scale or rotate it and get instant feedback in the form of artistic routes displayed in real time.

# Acknowledgements

I would like to express my gratitude to the individuals who have played an integral role in shaping the outcome of this project. First and foremost, I am immensely thankful for the support and guidance provided by my supervisors. Dr. Liangliang Nan has been an exceptional mentor, always there to offer reliable insights from a curious perspective. Similarly, Dr. Jantien Stoter's remarks and thought-provoking questions have led to significant improvements in my work.

I would also like to extend my appreciation to Dr. Michael Weinmann. His thorough review and remarks helped me refine the structure and the content of this thesis.

Additionally, I want to express my thanks to my girlfriend, Sylwia. Her sense of aesthetics served as guidance for some visual aspects of this thesis.

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# Acronyms

| | | |
|---|---|---|
| **GPS** | Global Positioning System | 1 |
| **CRS** | Coordinate Reference System | 5 |
| **RTMV** | Relative total metric value | xi |
| **LPIPS** | Learned Perceptual Image Patch Similarity | 28 |
| **WKT** | Well-known text | 11 |
| **GPX** | GPS Exchange Format | 13 |

# 1 Introduction

In this chapter basic concepts in the scope of Global Positioning System (GPS) art are explained. Related scientific work is reviewed and the research questions are formulated. Chapter 2 outlines the problems to be solved and the techniques that will be used. Chapter 3 explains the engineering decisions and the details of implementation supported by initial result visualizations to illustrate some ideas. Chapter 4 contains the results of testing the developed solutions, along with a summary and discussion of advantages and limitations. Chapter 5 presents the final conclusions and directions for future research.

## 1.1 What is GPS art?

GPS art is a term used to describe drawings created as a result of subsequent changes of one's location while tracking the movement, for example using device with a GPS receiver. If executed with a specific plan in mind, the recorded route can become an elaborate artwork (see Figure 1.1).

People create GPS art to make their movement routines more interesting. This is relevant to the social aspect of movement tracking apps (Runtastic[1], RunKeeper[2], Strava[3]) which encourage their users to share their activities. In this sense, the creative routes provide a bigger value, since they are more appealing to the eye compared to a standard route without any recognizable pattern. Making artistic routes is an additional motivation for people to engage in physical activity, which can have a positive impact on health. Strava, the provider of a widely used mobile app for tracking physical exercise, has created an online hub[4] for GPS art, where people can display their creativity and draw their inspirations. The GPS art creator community consists mainly, but not exclusively, of hikers, runners and cyclists. However, this is just the most common target group and not in any way a limitation. As an example, there exist some GPS artworks created by plane pilots[5] which just proves that in this kind of activity all that matters is the continuous change of position and not the means to achieve it. For the purpose of this thesis, we will be considering only the ground-based movement activities.

Creating GPS art of your own is a cumbersome process. It all starts with an idea of what a person wants to sketch on the map. The route plan has to consider various aspects. Since the movement is done on the ground, the movement patterns are constrained by the specifics of the surrounding environment, like road network layout and physical obstacles. Manual planning of an artistic route requires a person to carefully study the roads in the concerned

[1] https://www.runtastic.com/

[2] https://runkeeper.com/cms/

[3] https://www.strava.com/

[4] Strava art: https://www.strav.art/home

[5] Plane GPS art:
https://www.huffpost.com/entry/flightradar24-plane-drawings_n_56e69bbce4b065e2e3d65fae

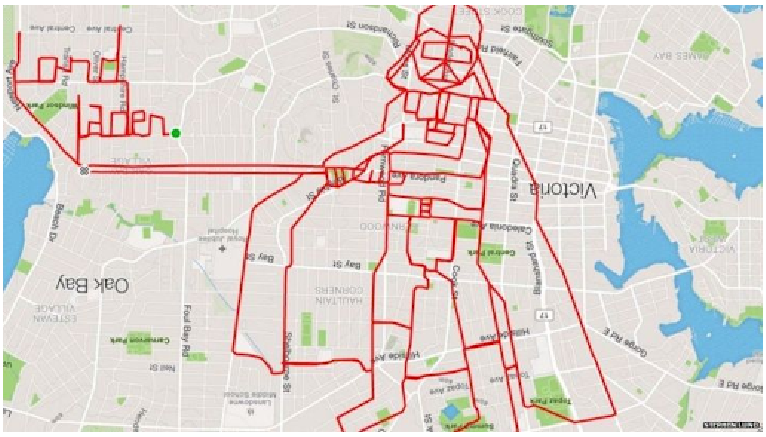


Figure 1.1: Darth Vader drawn as GPS art. (link)

area and is therefore not a quick and simple process. An example article[6] describes the efforts of GPS artists at work. In practice, it involves importing a relevant section of a road network map to a sketching tool and then sketching along the roads to obtain a route representing what the artist had in mind. Such a route would then need to be verified in terms of validity (traffic laws) and any potential false assumptions (for example roads connecting in a different way than the artist had imagined while designing).

## 1.2 Related work

### 1.2.1 GPS art generation

There is relatively little documented scientific research in the specific field of GPS art generation and none of the publicly available results give a full solution to the problem of making a GPS art route based on any input drawing.

Balduz [2017] describes a raster-based approach to generating GPS art. His idea was to transform both the input drawing and a road network into binary images, where the lines (street or drawing) are represented by ones and everything else as zeros. Then, an exhaustive search algorithm was supposed to find the best possible placement of the drawing by placing it in the position where the sum of the distances between the input drawing and the road network was the lowest. The disadvantage of such a solution is that it merely works on the rasterized version of the road network data and therefore cannot take into account the road network topology and traffic rules. This can result in an invalid route, for example in cases where the lines connect in the road network image, but in the actual road network data the connection does not exist or it is forbidden by the traffic laws.

Waschk and Krüger [2018] propose a more versatile method which solves the transformation-fixed (see Section 2.1) problem with a graph-based solution. In their work, they describe a workflow which generates artistic routes using the Dijkstra algorithm and their own custom cost function. Their approach assumes that the drawing is a single connected set of lines

[6] Designing artistic routes: https://www.runnersworld.com/runners-stories/a32433537/strava-art/

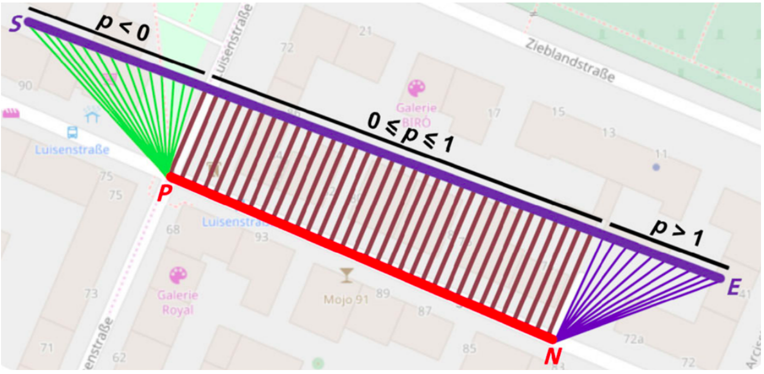


Figure 1.2: Metric C3, used to measure the distance between a street segment and a drawing segment. Source: Waschk and Krüger [2018]

with a predetermined starting point expressed in real-world coordinates. The algorithm visits the drawing segments one by one, and for each segment it routes from its start point to its end point. The end point of the previously visited segment then becomes the start point of the next one, which means that the resulting GPS art route will be a valid, connected route. The routing for each segment uses a cost function composed of different metrics defined in a way that the final route should resemble the input drawing as much as possible. The metrics used in the edge cost function are as follows:

- C1: distance from the edge end point to the final destination node
- C2: length of the edge
- C3: distance between the edge and the currently considered input drawing segment

While the first two metrics are straightforward, the third metric can have many interpretations, because there are many ways to measure a distance between two lines. To measure the distance between a given graph edge and a drawing segment, Waschk and Krüger [2018] propose a function that calculates a sum of distances from a number of points, spaced evenly on the drawing segment, to the graph edge (see Figure 1.2). The details of how many points should be used for the C3 metric are omitted. This metric is further explored in the course of this thesis (Section 3.1.1).

The paper by Waschk and Krüger [2018] does not describe what weights should be assigned to the metrics and how changing them impacts the artistic routes. Another unresolved problem is the transformation-agnostic approach (described in Section 2.1), since their solution assumes that the user should define the coordinates of the starting point of the GPS art route.

The mentioned research serves as a good theoretical foundation and a starting point for further improvements in the course of this thesis.

### 1.2.2 Quality assessment of artistic routes

Zhang [2021] describes the implementation of a Convolutional Neural Network for the purpose of doodle sketch recognition. Although implementing a classifier for human-made drawings is out of scope of this thesis, some ideas may be useful when using existing machine learning models for the evaluation of the quality of the results. The GPS art routes are mostly based on a simplistic set of lines, since the real-life road network layout rarely enables creating something complex. For this reason, a doodle sketch dataset can be a good reference base for comparisons with GPS art routes. The certainty with which the machine learning model evaluates the resulting artistic routes can be used as a fuzzy measure of its quality.

Another approach for quality evaluation is to use perceptual similarity algorithms. Although implementing such an algorithm is out of scope of this thesis, there are sources which can be used in our work. Zhang et al. [2018] describe an algorithm which uses deep features as a metric of perceptual similarity between images. Such a metric can be calculated from the source drawing and the output artistic route to determine how perceptually similar they are.

## 1.3 Research questions

In this thesis, we will explore ways to automate the process of designing a GPS art route from any drawing. The aim is to translate the user input (for example contour lines) to a valid route in the road network, while retaining the intended semantic meaning. Additional user requirements can also be considered, for example a desired starting point or a desired route length. A solution for this problem could be valuable for the developers of physical activity tracking apps widely available on the market.

The main research question can be defined as:

**How to automatically generate artistic routes based on any input drawing?**

The resulting sub-questions are as follows:

- How to define measures of quality and how to evaluate them for an obtained GPS art route?
- What priorities / compromises should the designed algorithm have in order to produce optimal output considering the user's preferences?

The scope of research will include designing, implementing and testing an automated workflow, capable of producing artistic routes from source drawings. To improve the workflow and tune its parameters, a framework for quality evaluation of the results will also be implemented.

# 2 Methodology

In this chapter the research methodology is described. The research problems and the plan to solve them is presented.

## 2.1 Problem statement

There are two separate approaches in solving the GPS art problem which are based on these two assumptions for the source drawing:

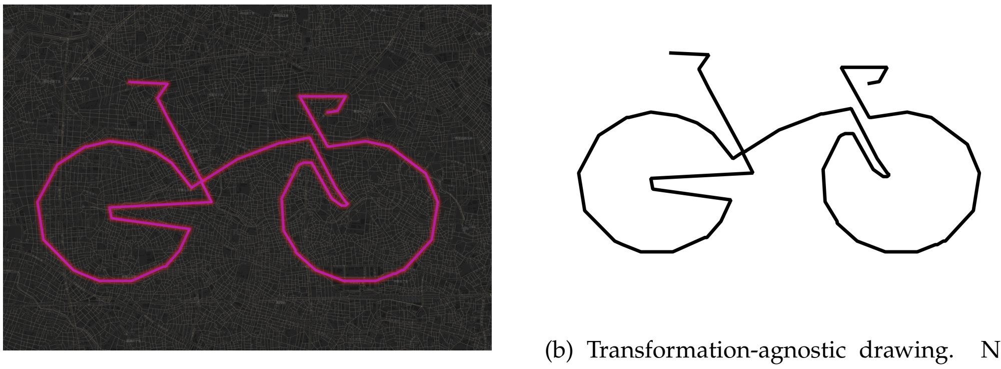
- transformation-fixed input
- transformation-agnostic input

The transformation-fixed (Figure 2.1a) input refers to an input drawing which has real-world coordinates and can be placed on the map as is, with no geometric transformation required. Such a drawing can be obtained by sketching with Coordinate Reference System (CRS) aware software on top of a map overlay of the area of interest. The solution then is to use a standard routing algorithm in combination with a custom edge cost function to optimize the route for maximum similarity to the source drawing. The method described by Waschk and Krüger [2018] is a good starting point for further improvements.

The transformation-agnostic (Figure 2.1b) input assumes that the real-world position of the drawing is unknown initially. This could mean that the drawing was simply drawn on a blank piece of paper and has no relation to real-world coordinates. The problem then becomes more complex, since the transformation properties of the drawing have yet to be determined. The drawing would have to be subject to an affine transformation to spatially relate to the road network in the area of interest. To achieve this, an image matching algorithm can be used to search for patterns in the road network and select a potential fixed location, size and rotation of the drawing. Once that is done, the problem becomes transformation-fixed and the solution is as described for the transformation-fixed input.

Solving the transformation-agnostic approach has some significant obstacles. One of them is the fact that there are infinite possibilities of transforming the input shape and countless ways of representing it using a real-world route. Another issue is that for any given input drawing and the obtained artistic route there is no clear reference solution. The standard approach would be to verify the outputs of the algorithm by comparing it to a manually prepared reference solution. However, having a complex route network and an artistic route calculated by the algorithm, it is a challenge to manually confirm whether this particular route is the globally optimal solution.

Additionally, looking at the GPS art algorithm from a user's perspective introduces some extra demands for the algorithm. Proximity to a chosen starting location (e.g., the user's home from where he starts the hike/jog) or a specific route length (e.g., a half-marathon) are important requirements for a real-world use case. These extra constraints require some



(a) Transformation-fixed drawing overlaid on top of a road network map.

(b) Transformation-agnostic drawing. Note that this kind of a drawing cannot be overlaid on a map, since it has no CRS.

Figure 2.1: Two kinds of input to the GPS art problem

compromises in the algorithm, thus making the solution deviate even more from the global optimum.

In the course of this thesis, solutions for both the transformation-fixed and the transformation-agnostic approach will be presented. Finally, the results will be combined into a single workflow for automatic artistic route generation. As an alternative solution, an interactive GPS art application will be developed as well. Its purpose is to generate artistic routes with real-time feedback for the user.

### 2.1.1 Template matching

Template matching is a known method for detecting the location of a certain pattern (template) in a bigger image. One idea for solving the transformation-agnostic approach is to convert the input drawing and the road network in the area of interest to a raster format (e.g., GeoTIFF[1]). Such files can then be used as inputs to a template matching algorithm where the input drawing is considered the template and the road network is the larger image where we expect to find it. The expected result is that the drawing can be placed on the map in a location where it matches the road network in the optimal way in accordance with the criteria used in the template matching algorithm. A significant challenge comes from the fact that there is little probability of a perfect match existing in the road network, because of the irregular pattern of the street segments. Knowing this, the aim is to find the most matching location, even if it is not really a close match.

Template matching is a well-researched problem in computer vision. Many techniques have been developed over the years to solve complex problems like object detection, facial recognition, pattern detection. Two main approaches can be listed as a high-level overview of the existing techniques:

- Template-based
- Feature-based

[1] https://en.wikipedia.org/wiki/GeoTIFF

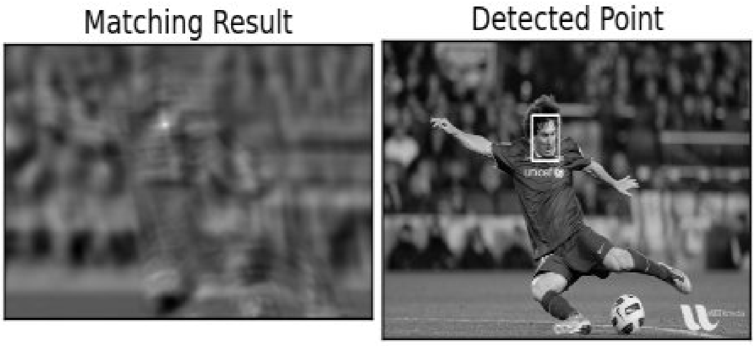


Figure 2.2: Example template matching solution to detect a person's face. Source: OpenCV tutorial

The most basic form of template matching relies on the use of known templates. It searches for the optimal location of the template by sliding it along the entirety of the target image and quantifying in each location the difference between the intensity (pixel) values of the template and the searched image (Brunelli [2008]).

Instead, feature-based approach uses features like shapes, textures, corners, colors. The features can be extracted using deep neural networks (example research by Lei et al. [2021]). This approach can make use of rotation-invariant and scale-invariant feature descriptors to find a solution across multiple scales and rotations. On the other hand, it is more difficult to apply this to the GPS art case. This is because most of the feature descriptors are designed to work on texture-rich images with either RGB or at least grey-scale color information. Images of real life objects usually satisfy these criteria (see Figure 2.2), whereas binary images representing sets of line segments do not. The lack of diverse pixel intensity means that the feature descriptor algorithms have problems detecting distinct key points and the matching result can be irrelevant.

In the scope of this thesis, we focused on the simpler template-based approach. The details of the experiments are presented in Section 3.2.2. The main idea is to use a template matching algorithm to compare two grey-scale images of the input drawing and the target road network. The template is therefore a binary image with positive values representing the drawing's lines and zero values being everywhere else. The goal is to match the input drawing lines with the road network street segments. For this reason, when quantifying the difference between the template and the target, an important point is to use a mask made up of positive values in the template. This means that when sliding the template along the target image, the difference in each position is calculated considering only the pixels that make up the lines of the input drawing or road segments. Without using such a mask the proposed template matching method would not work correctly, because it would match using also the empty (zero values) spaces in the input drawing with the empty spaces in the target road network.

Figure 2.3 demonstrates the relevance of this idea to the GPS art use case with several examples. The simplest case is presented in Figure 2.3a. In this case the template is a small cross (red) and the target image is the big cross (white). If considered in a single-scale setup, this case has only one best solution, with the template cross being placed in the center. If considered in multi-scale set up, this particular case has infinitely many solutions, because the cross can be scaled to any size and still have a perfect match with the target image in its center.

The next example (Figure 2.3b) presents a small square (red) as the template and an unbounded tic-tac-toe board (3x3 square board without a boundary) as the target image. In this case, there exists only one best solution across all possible (aspect ratio locked) scales, and that is in the middle of the board, where the scaled template can have 100% overlap with the target image. This is the ideal case because it introduces no ambiguity and allows for a single choice of the location of the template. The template-based template matching approach can be augmented by generating multi-scale pyramids of the template and running the algorithm multiple times to obtain the best match across multiple scales.

Another example (Figure 2.3c) illustrates the case when no single solution can be found. The template here is a square (red) and the target is a bounded tic-tac-toe board (3x3 square board with a boundary). There are 9 possible solutions in this case and no way of determining which one is the best, since all the 9 potential positions of the template will have 100% overlap with the target image.

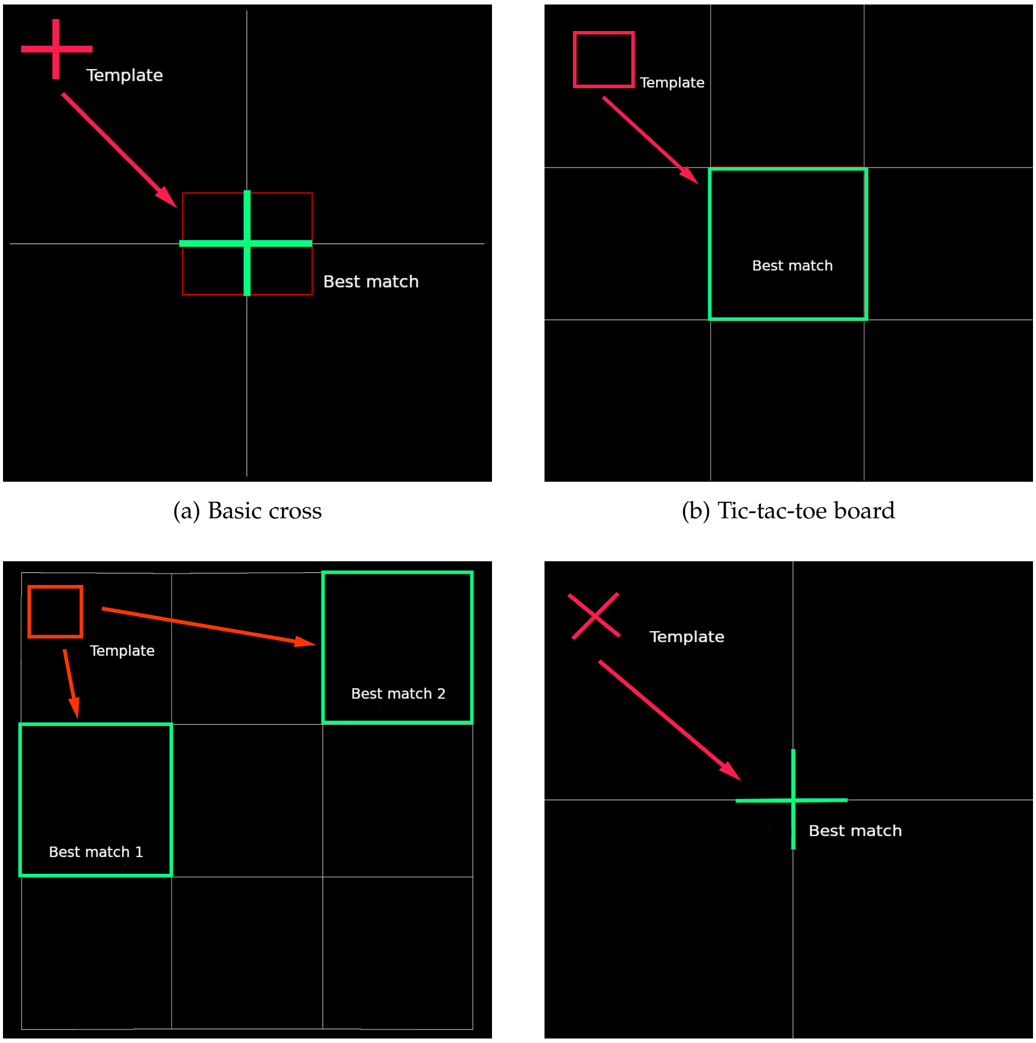
Another challenge of template matching is the relative rotation between template and the target. Cases like these cannot be handled when using the template-based template matching in its most basic form. An example is given in Figure 2.3d. The template here is a cross rotated by 45°. The template-based template matching method is not rotation-invariant. To solve this specific case with this method, it would require generating multiple variations of the rotated template and running the algorithm multiple times. This approach still does not guarantee that the optimal template position will be found, because it is possible that none of the rotated variants matched the exact rotation of the template in the target image.

#### Comparison with other methods

We chose template matching as a method for pattern recognition. The advantage of template matching in the GPS art case is that its result can be clearly explained as the position where the overlap between the input shape and the road network is the highest. There are also other ways to approach this problem.

One alternative approach to the problem is to use deformable shape matching. Kim and Shontz [2010] propose a method which represents a shape as a triangular mesh and uses an energy function to compare the source and the target shape. Their solution is invariant to rotation, translation and uniform scaling. However, such a method is not easily applicable for the GPS art case. To begin with, it operates on closed shapes with a clearly defined interior, a pre-condition that is not always met by single-line drawings. Figure 2.4 shows an example of a shape that is not closed and does not have an interior. On top of that, the distinct shapes have to be pre-defined, so that the source shape can be compared to them to select the best match. This is not the case with a street network image, where there are no clearly distinguishable shapes, and the detection of any candidate shapes is a separate problem on its own.

An alternative way is to skip the image-based technique and try exhaustive search with the transformation-fixed approach, by iteratively transforming (scale, translate, rotate, shear) the input and running the routing algorithm multiple times to select the optimal GPS art route by trial and error. This idea is further explored in Chapter 3.



(a) Basic cross

(b) Tic-tac-toe board

(c) Ambiguous case

(d) Rotated cross

Figure 2.3: Different basic examples of template matching

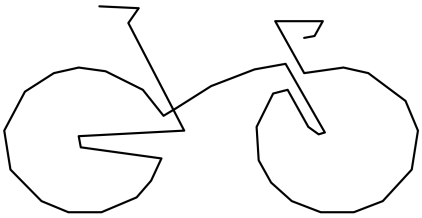


Figure 2.4: Example of a single-stroke drawing with no clearly-defined interior

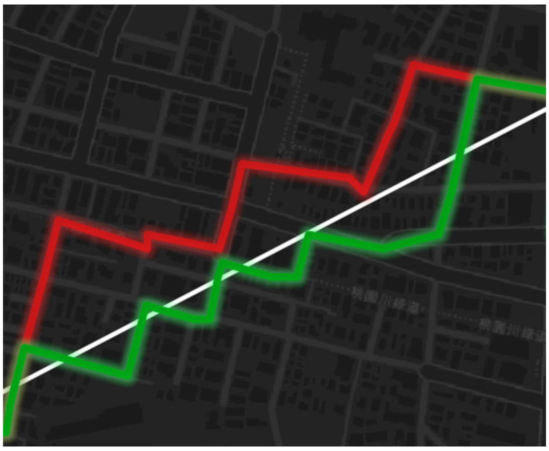


Figure 2.5: Examples of different routing results for a given segment of the input drawing (white). The red route is the shortest one. The green route is optimized to be as close to the drawing segment as possible.

### 2.1.2 Cost function

In the routing part of the solution, the A* algorithm is used for each segment of the input drawing. The road network is loaded as a connected graph consisting of edges (road segments) and nodes (start points and end points of road segments). Each edge is given a cost and the algorithm computes the least expensive path from the start of the currently processed drawing segment to the end of this given segment. Standard routing solutions define the cost as the length of an edge or the time it takes to traverse an edge. However, GPS art routing has a different goal and therefore should use a different cost definition which should be dependent on the input drawing. For this purpose, a custom cost function is developed, which aims to make a route as close to the input segment as possible, while also still reaching the target node. The resulting route is therefore not necessarily the shortest one (see Figure 2.5). The custom cost function can be composed of several metrics, for example based on graph edge to drawing segment distance. The metrics used in our experiments are described in detail in Section 3.1.1.

### 2.1.3 Evaluation

Having metrics for measuring the quality of results will be crucial in tuning the final solution. Since the solution will be algorithmic, there will be ways to obtain a numeric evaluation of the resulting artistic routes. However, taking into account the artistic aspect, evaluating the aesthetic value of a resulting route is also important. One way of achieving this could be to use a machine learning algorithm trained to recognize human-made drawings. If the algorithm can recognize with high certainty the semantic meaning of a given GPS art route, then it can be assumed that the result can also be meaningful to human eyes. The implementation of a machine learning model for this purpose is out of scope of this thesis.

Another idea for measuring the quality of the resulting GPS art is to compare it with the input drawing using a perceptual similarity metric proposed by Zhang et al. [2018]. In this approach, we try to measure the perceptual loss between the resulting GPS art and the source drawing, using a method that is meant to resemble human perception. Implementation of a perceptual similarity metric is out of the scope of this thesis. Zhang et al. [2018] share their source code[2] for the perceptual distance function.

The overall assessment of quality will be composed of:

- Total sum of the metrics used in the routing algorithm cost function
- Results of machine learning object recognition for the resulting GPS art
- Perceptual loss distance between the input image and the output GPS art

### 2.1.4 Automatic GPS art workflow

The automatic GPS art workflow is based on the multiple techniques and solutions developed in the course of this thesis. A combination of image-based and routing-based approaches is used to find an optimal artistic route considering a given input drawing and user-defined parameters. The workflow is summarized with the diagram in Figure 2.6.

The user can specify:

- input shape in a vector format (e.g., Well-known text (WKT) format)
- area of interest (e.g., a certain city district)
- starting point of the route (e.g., the user's home) and an allowed threshold (e.g., no more than 150m from the user's home)
- desired route length (e.g., a half-marathon)

The area of interest and starting point parameters are mutually exclusive. Only one of them can be used at a time. Real-life use cases require the possibility of defining extra requirements for the route. For example, a user may want to run an artistic route half-marathon (21 kilometers in length), starting from a location within 150 meters of their home. Alternatively, they may accept a route that does not start at their home, but is still located within a certain neighborhood within a desired area of interest. Requirements like these introduce some constraints for the result of the algorithm and require a separate implementation.

The process starts by converting the vector format input drawing to an image format which can be used in the next steps. The template matching step attempts to find a location of the drawing where its overlap with the road network is the highest. After finding such a location, the transformation parameters for the input drawing are obtained. This transformation is used to shift the input vector drawing to the position determined by template matching. The shape is also simplified (see Section 3.4) before the next steps. With the shape in the starting position, the exhaustive search is run, calculating routes for many different transformations of the input. The numerous artistic route candidates are then cleaned in a postprocessing step (see Section 3.4). The number of possibilities depends on the number of different transformations defined. The hundreds or thousands of candidates get initially filtered using RTMV (explained in 3.3.1) to narrow down the choices. It is the fastest way to efficiently filter out the results that are of low quality. The RTMV, which is a metric of geometric distortion from the source shape, cannot be used by itself to select the best route candidates but it can be used to filter out the worst ones. The remaining candidates

[2] Github repo: `https://github.com/richzhang/PerceptualSimilarity`

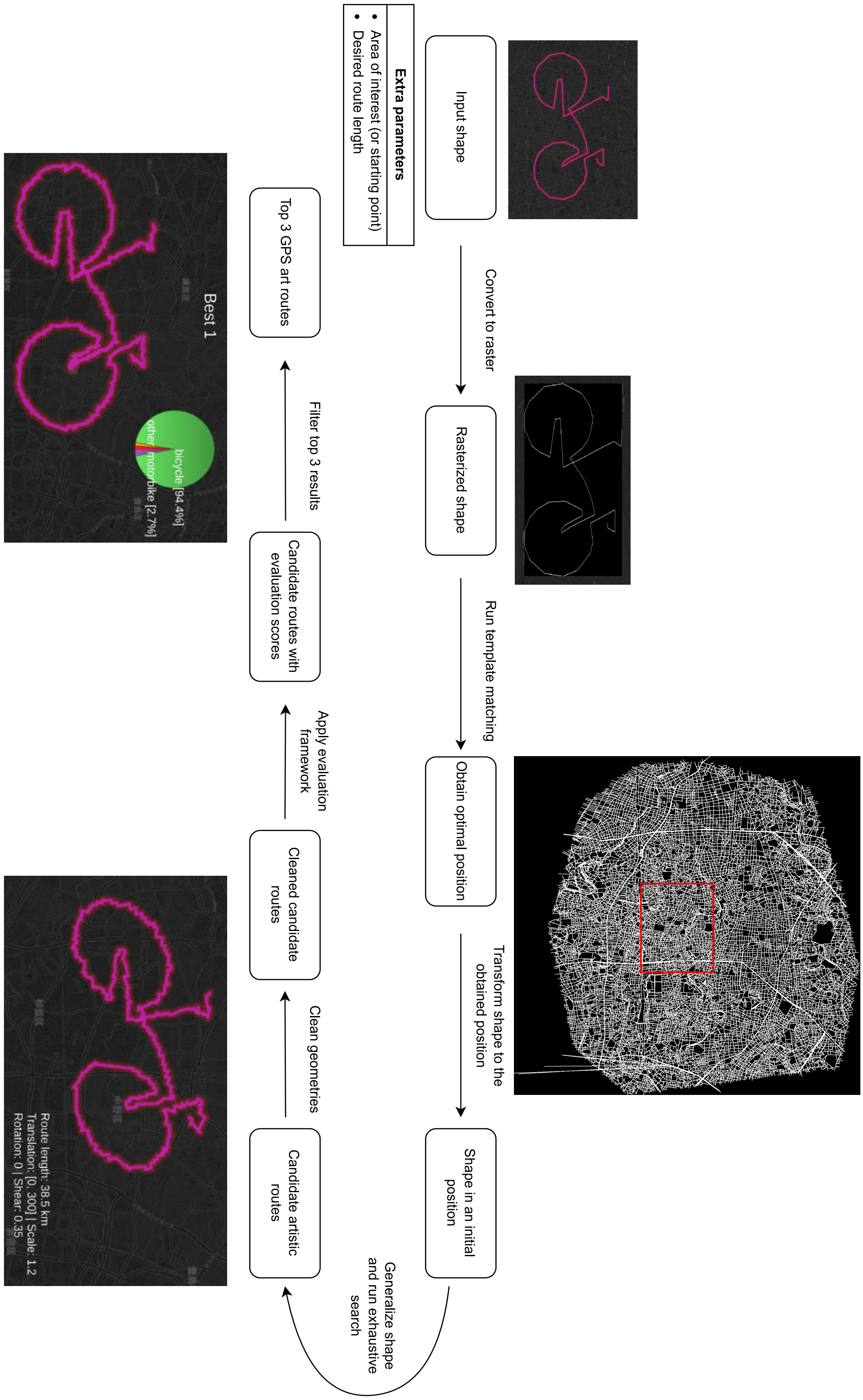

Input shape
Extra parameters
Area of interest (or starting point)
Desired route length
Convert to raster
Rasterized shape
Run template matching
Obtain optimal position
Transform shape to the obtained position
Shape in an initial position
Generalize shape and run exhaustive search
Candidate artistic routes
Route length: 38.5 km
Translation: [0, 300] | Scale: 1.2
Rotation: 0 | Shear: 0.35
Clean geometries
Cleaned candidate routes
Apply evaluation framework
Candidate routes with evaluation scores
Filter top 3 results
Top 3 GPS art routes
Best 1
bicycle [94.4%]
motorbike [2.7%]
other

Figure 2.6: Automatic GPS art workflow diagram

are evaluated using the complete evaluation framework consisting of RTMV and the machine learning based methods (described in Section 3.3.2). The final score is calculated as a weighted average of the RTMV, object recognition certainty and perception loss distance. The top-3 results with the best score are presented to the user who can then make a visual assessment himself, pick a route and export it to a GPS Exchange Format (GPX) file.

Specific implementation aspects of the automatic GPS art workflow are described in Chapter 3.

### 2.1.5 Interactive web application

To integrate interactive user inputs, a GPS art web application with a graphical interface will be developed. Some of its functionality will be based on the same solutions as in the automatic GPS art workflow (Section 2.1.4). The main idea is to rely on the user's intuition for the initial placement of the input shape on the map. The user will be given the possibility to load his own input shape, apply affine transformations (via the use of clickable controls) and view the artistic routes as they are generated in real-time. To allow integration with modern activity tracking apps, the export of a route to GPX will also be possible. The application should be tuned for high responsiveness to be enticing for its users.

# 3 Implementation details

This chapter provides an overview of various implementation aspects of the chosen approach. The solutions to the problems presented in Chapter 2 are presented first as separate components and then in the form of complete products. Some implementation ideas are supported with preliminary visualizations of the results.

## 3.1 Transformation-fixed approach

### 3.1.1 Cost function

The core of the solution to a transformation-fixed problem is based on the algorithm presented by Waschk and Krüger [2018]. The idea is to iterate each segment of the input drawing and perform routing with a custom cost function from its start point to its end point. The end point of the previously visited segment becomes the starting point for the next one. Repeating the process for all input drawing segments allows us to obtain a single connected route. This solution assumes that the drawing is a single connected set of lines with a predetermined starting point expressed in real-world coordinates.

The following pseudo code represents the algorithm used for generating artistic routes:

```
Algorithm 3.1: Generate_GPS_art (D, G)
Input: D: input drawing as an ordered set of line segments, G: road network graph
Output: Path in the road network approximating the input drawing
foreach segment $\overline{AB}_i \in D$ do
    $S_i, E_i \leftarrow$ get nodes of graph $G$ which are the closest to the start and end of $\overline{AB}_i$
    $Route_i \leftarrow$ get route from $S_i$ to $E_i$ in graph $G$ using the A* algorithm with custom
      cost function
$Result \leftarrow$ connect the paths for all the segments to make a single connected route
```

The custom cost function used in each routing algorithm run is meant to optimize the route to match the drawing segment. As a result, the final artistic route should resemble the input drawing as much as possible in accordance with the defined similarity metrics. Making this algorithm work for the GPS art case requires defining the cost function and selecting the weights for its components.

The version of the cost function proposed in this thesis uses two metrics which constitute the cost of a given edge in the context of the currently routed drawing segment:

- C2: Edge length

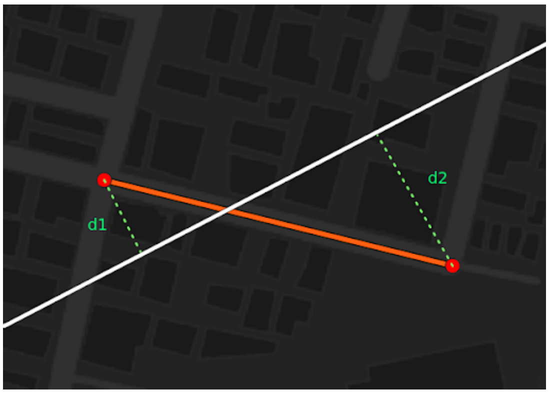


Figure 3.1: A visualization of the metric function. Two shortest distances d1 and d2 (green) are calculated from the end points of the graph edge (orange) to the drawing segment (white). The background shows the street network for reference.

- C3: Similarity metric between the graph edge and the current drawing segment

The metric C1, proposed by Waschk and Krüger [2018], has not been included in our version of the cost function. Its value comes from the euclidean distance between the current graph edge end point and the drawing segment endpoint. The researchers justify its choice by the need to minimize the distance to the route's end. However, the path length minimization is already provided by the metric C2, which results in finding the shortest path if used as the only component of the cost function. The only other requirement for the route, except being as short as possible, is that it also has to be spatially close to the drawing segment. This is satisfied by the C3 metric. The C1 metric brings no additional value, while adding more computational overhead, and was therefore removed. The aspect of directing the route towards the final goal is solved by using A* algorithm with an euclidean distance heuristic function (described in Section 3.1.2).

In most common routing scenarios we would only be interested in using either an edge length or the time it takes to go from its start point to its end point. With the cost of an edge defined in this way, the routing algorithm optimizes for the shortest or the fastest path. In the case of GPS art, the aim is to have the path run as close to the input drawing as possible.

As mentioned earlier, the routing algorithm is run for each segment of the input drawing. The similarity metric is a function dependent on the graph edge and the input drawing segment. It can be defined as:

$$M(S_g, E_g, S_d, E_d) = \frac{distance(S_g, \overline{S_d E_d}) + distance(E_g, \overline{S_d E_d})}{2}$$

where:
$S_g$ - graph edge start point
$E_g$ - graph edge end point
$S_d$ - drawing segment start point

$E_d$ - drawing segment end point

The entire cost function can be defined as:

$$C(S_g, E_g, S_d, E_d) = \alpha * distance(S_g, E_g) + \beta * M(S_g, E_g, S_d, E_d)$$

where:
$\alpha$ - weight of the edge length
$\beta$ - weight of the similarity metric

The C3 metric has been revised in this thesis. Its value is based on the knowledge that the two compared lines (graph edge, drawing segment) are always simple segments. It is therefore not necessary to deploy complex curve comparison methods like Hausdorff distance or Fréchet distance. The metric function calculates the average of the shortest distances from the end points of the graph edge (road network segment) to the drawing segment. This is illustrated in figure 3.1. Comparing it to the similarity metric proposed by Waschk and Krüger [2018], the metric introduced in this thesis is simpler and needs only two distance calculations (from the street segment end points) instead of numerous calculations for many points.

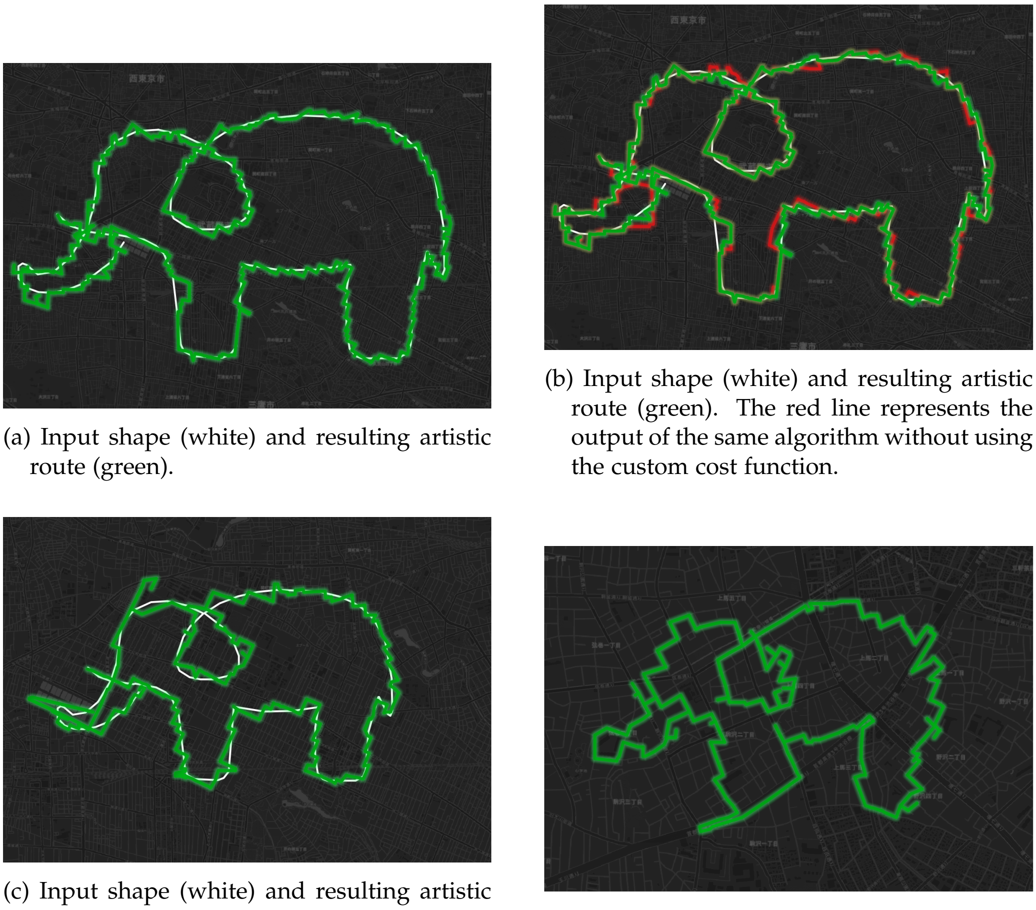
Using the Algorithm 3.1 with a custom cost function allows us to obtain a route approximating the input drawing (as shown in Figure 3.2). Figure 3.2a shows the resulting artistic route overlaid on top of the input drawing (white). Figure 3.2b also shows what the route would look like if it was optimized for the shortest distance for each routed segment. Without specific metrics for segment closeness, the shortest route can often deviate significantly from the source drawing segments and is therefore not a correct solution to the GPS art problem.

It should be mentioned that the local layout of the road network, as well as a relatively smaller scale of the input drawing, can be significant limiting factors for the obtained artistic route solution. The route presented in Figure 3.2a has a total length of 46 kilometers. Scaling down the input shape or moving it to a place with less dense street network will at some point result in a lower aesthetic value of the generated artistic route. The example route in Figure 3.2c has a length of 24 kilometers and already has significant visual obstacles like road segments significantly deviating from the source drawing. Scaling down even more to a 10 kilometer route (Figure 3.2d), the approach cannot approximate the input drawing with a reasonable street graph anymore and it is impossible to recognize what the artistic route is meant to represent. A similar problem can occur even without down-scaling the input shape, but just simply by changing its position. The resulting artistic route is highly dependent on the characteristics of the underlying street network. In general, a location with a dense street network in the area is more likely to produce a better artistic route output than a location with fewer streets. This does not always apply, because the input drawing can be any kind of shape. For example, a certain area of interest with a sparse street network can have its road segments laid out in a way that suits this particular shape better than some other area despite having more street segments.

### 3.1.2 Choice of the routing algorithm

Waschk and Krüger [2018] propose the usage of Dijkstra's algorithm[1] for the routing tasks.

[1] https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm



(a) Input shape (white) and resulting artistic route (green).

(b) Input shape (white) and resulting artistic route (green). The red line represents the output of the same algorithm without using the custom cost function.

(c) Input shape (white) and resulting artistic route (green). The smaller scale of the input makes the generated route less visually similar to the source.

(d) Resulting artistic route (green) for an even smaller scale of the input. The visual quality is degraded.

Figure 3.2: Different examples of artistic routes of an elephant obtained using Algorithm 3.1. Green color represents the route and white color represents the input drawing.

As a slight performance improvement, the Dijkstra's algorithm was replaced with the A* algorithm[2]. The principle of the solution remained unchanged, but the performance increased slightly, in some cases up to 15%. This aspect has not been benchmarked in this thesis, since the performance gain from switching Dijkstra's algorithm to A* for certain scenarios has been proven in many publications over the years. Example research by Rachmawati and Gustin [2020] or Santoso et al. [2010] present the advantages of using A* in road network routing problems. It is proven that the A* algorithm is more efficient in cases when the route search needs a fuzzy direction (e.g., towards the target node), not to explore the nodes that have no chance of being in the final route (e.g., nodes in the opposite direction from the target node). In road network graphs, the A* algorithm with an euclidean distance heuristic function allows the target point to be reached in fewer steps over the graph nodes.

The result of routing from node A to node B is guaranteed to be identical as in Dijkstra's algorithm as long as the heuristic function is admissible. Knowing the cost function (explained in Section 3.1.1) as:

$$C(S_g, E_g, S_d, E_d) = \alpha * distance(S_g, E_g) + \beta * M(S_g, E_g, S_d, E_d)$$

We can define the heuristic function as:

$$H(N_i, N_e) = \alpha * distance(N_i, N_e)$$

where:
$N_i$ - currently visited graph node in the A* search
$N_e$ - target graph node (route end)
$\alpha$ - weight of the edge length in the cost function

This way, it is certain that the heuristic will never overestimate the cost, therefore satisfying the admissibility criterion. At the same time, it will give the A* search a direction heading towards the route's goal.

## 3.2 Transformation-agnostic approach

### 3.2.1 Exhaustive transformation-fixed search

One approach for solving the transformation-agnostic approach is to use an exhaustive transformation-fixed approach. It is assumed that any input drawing can be expressed as an artistic route in many possible variations which can still preserve its semantic meaning or aesthetic value. This method starts with a source shape that is placed in an approximate position on the map, within a given area of interest. Then, combinations of affine transformations (see Figure 3.3) are used to warp this shape into many variations. Algorithm 3.1 is run for each of these transformed shapes, which results in many candidate artistic routes. These routes then need to be evaluated using some criteria which will allow to choose the

[2] http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html

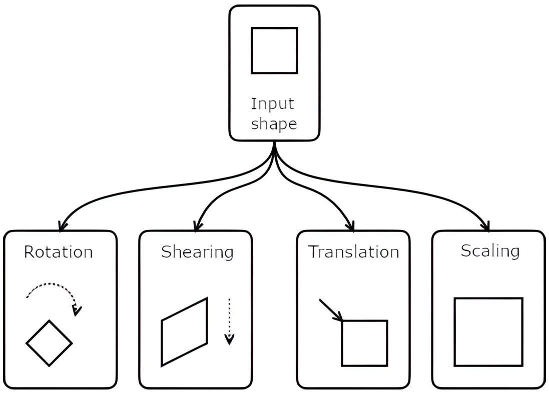


Figure 3.3: Affine transformations used to warp the input shape. Source: link

overall best result. Combining this method with an automatic scoring system described in Section 3.3 gives an effective method of choosing an optimal GPS art solution automatically.

The main advantage of this method is its effectiveness. It determines the best artistic route location by comparing the properties of actual route candidates (see Figure 3.4) based on road network data. This is important, because some route properties, like the length or its exact shape, become only available after the routing itself is done. For example, an image (raster) based solution like template matching has no way of telling the exact shape of the route, because it only works on image representations of the roads and not the actual road network graph.

The expected disadvantage of this exhaustive search method is its efficiency. While the routing algorithm itself is efficient, it has to be run hundreds to tens of thousands of times (depending on the granularity of the applied transformations) which is time-consuming. The quality of results is also dependent on the granularity of the different transformation variations. This introduces the risk of skipping a potentially good route candidate if the transformation variations are too sparse. For example, if the shape is only checked in one position every 200 meters, then the optimal position can be missed and not included in the final solution.

### 3.2.2 Template matching

An implementation of template-based template matching is available in the OpenCV library (Bradski [2000]). The quantification method for the difference between the template and the target can be one of the following as per the documentation[3]:

- Sum of squared differences (TM_SQDIFF)
- Cross correlation (TM_CCORR)
- Mean-shifted cross correlation (TM_CCOEFF)

[3] OpenCV Docs: https://docs.opencv.org

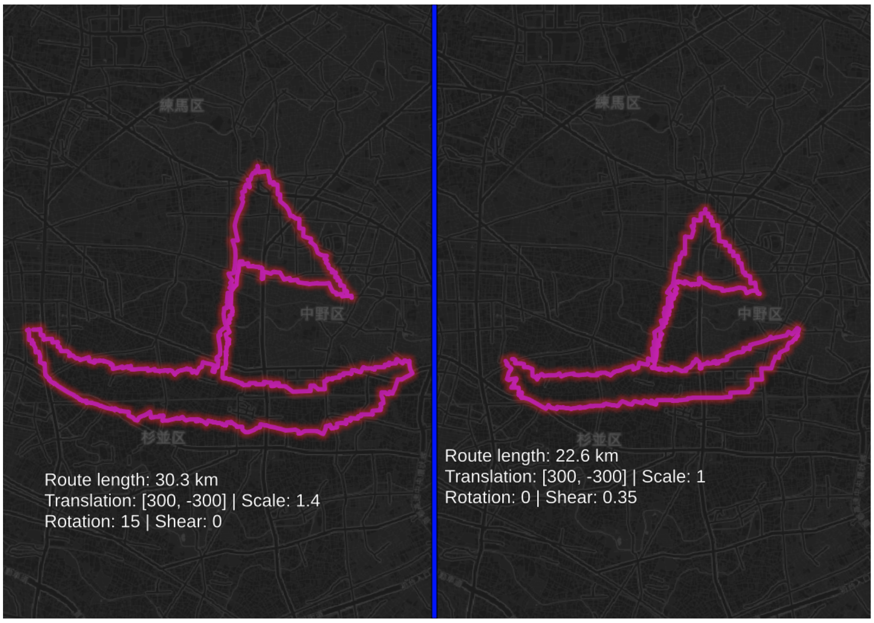


Figure 3.4: Example two GPS art route candidates for an input sailboat drawing. These particular candidates have different scale, rotation and shear

Additionally, normalized versions of these metrics are available, to be used for comparisons of results for templates in different scales. For the experiments in this thesis, we chose the sum of squared differences and used our own normalization formula to be able to compare the results for the templates in different scales and choose ultimately the best one. The formula relies on the prior knowledge that the input images are binary, so in the context of OpenCV workflow, all the values are either 0 or 255. This allows retrieving the metric that can be compared between the templates in all scales to determine the best one: the percentage of pixels of the input drawing that overlap with the road network. Figure 3.5 demonstrates the usage of template matching for the GPS art case. The determined location (red rectangle) is characterized by the highest overlap of the input image and the road network image. Figure 3.6 shows the same case from a more zoomed in perspective. It is visible that the sailboat is placed in a location where it has many overlapping road segments. If the drawing was moved slightly to the side, the central part of the ship would no longer have significant overlap with the long straight road segment (visible in the centre of the left picture in Figure 3.6) and the total overlap would be lower. It is generally a challenge to verify the correctness of template matching results for GPS art cases, because it is known that the perfect result does not exist. The overlap values are usually quite low, so the final choice is the best of many suboptimal options. In the presented case, the overlap is only around 35%. If there was a sailboat of this exact shape hidden somewhere in the street layout then the overlap in such a location would be equal to 100%.

This works with the assumption that the template has reasonable scale range. If it is made really small, it can easily have 100% overlap with the roads, although it will be too small to create meaningful GPS art out of it.

A disadvantage of this approach is that it does not give a complete solution. The result of template matching in this context is just a position of the drawing in the road network. To

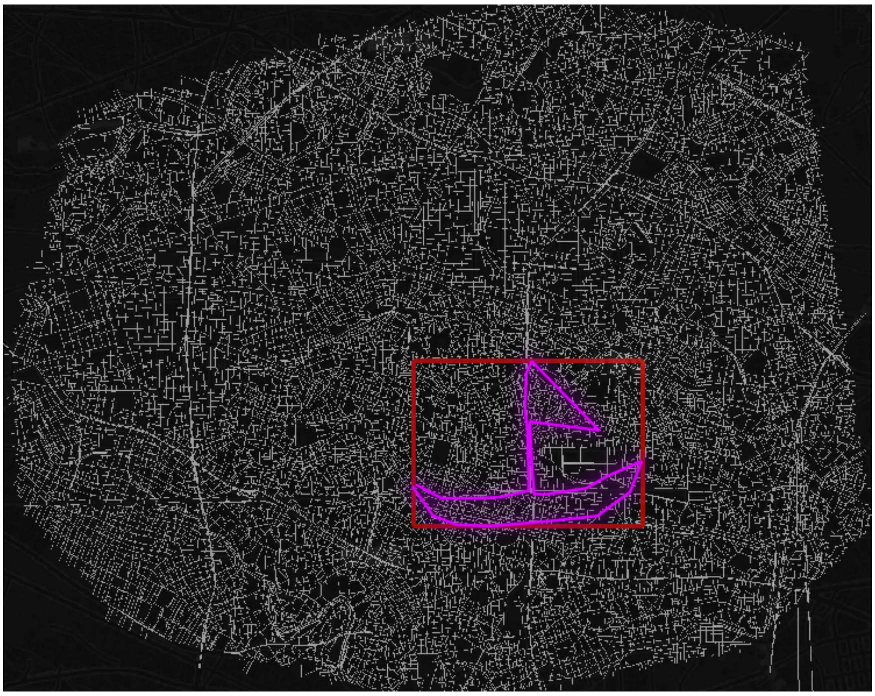


Figure 3.5: Example of a template matching result for a drawing of a boat. The red rectangle marks the area where the drawing overlaps the most with the road network.

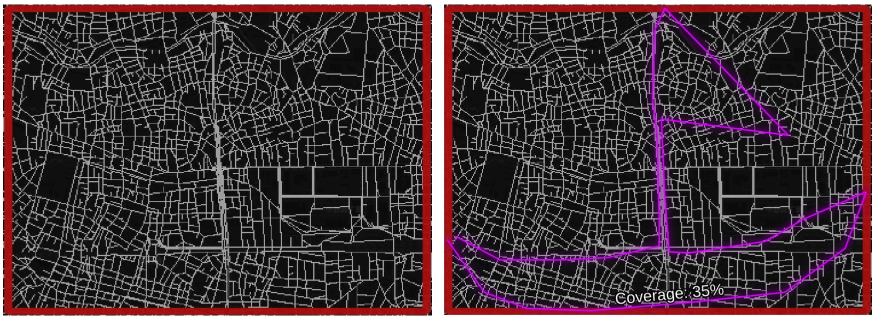


Figure 3.6: Closer look at the case illustrated in Figure 3.5. In this example 35% of the drawing image positive pixels overlap with the road network image positive pixels.

complete this, the GPS art routing algorithm still has to be run for the drawing placed in this position. Template matching is good for initial placement of the drawing where it overlaps the most with the roads - usually the area with the most dense street network. In reality, the best artistic route can be located somewhere else, since its final shape depends on the road layout and accessibility of some streets.

Another limitation is the fact that template matching does not support elastic matching, which could be desirable in the GPS art case. The template search only considers the template variations based on affine transformations and is therefore not suitable for non-rigid shape deformations.

## 3.3 Evaluation of route quality

One thing to discuss when assessing multiple artistic route candidates for the same input is normalization for fair comparison. If the input to the algorithm was transformed in any way (translated, scaled or rotated - see Section 3.2.1) then for an objective evaluation its resulting GPS art candidates should be transformed using the reverse of this transformation.

While translation and scale (up to some extent) mostly do not impede a person's perception of an artistic route, one could argue that rotation and shear introduce an impediment in recognizing its meaning (see Figure 3.8). As an example, a person can easily recognize a rabbit in a drawing even if the rabbit is moved or changes its size (maintaining its length to width ratio and within reasonable scale). However, it is undeniable that a person can have a harder time recognizing a drawing of a rabbit that is presented upside down. Ultimately, the rotation of the GPS art is irrelevant, because the map can be presented in any arbitrary orientation. While the assumption of the north direction making up the vertical axis is the most common, in reality the map in the final visualization can be rotated to match the rotation of the artistic route. An adjustable axis orientation is a feature of most modern applications dealing with interactive maps. Figure 3.7 shows an example of such orientation adjustment using Strava[4].

To include the impact of shear on the perception of the GPS art, the shear will not be taken into consideration when reversing the transformation for the purpose of evaluation.

Section 3.3.1 describes a metric to evaluate geometric distortion of an artistic route relative to its source drawing. Section 3.3.2 describes learning-based methods of GPS art evaluation.

### 3.3.1 Accumulated cost metric

The previously mentioned numerical way of evaluating the quality of the artistic routes is explained below. The main idea behind it, is that if the routing algorithm uses a cost function to optimize the routing locally for each segment of the input drawing, then the best possible route should have the lowest cost in total. This simple assumption needs to be improved in order to be able to compare the quality of artistic routes based on inputs in varying scales.

It is known that the cost function for a single graph edge consists of:

[4] `https://www.strava.com/`

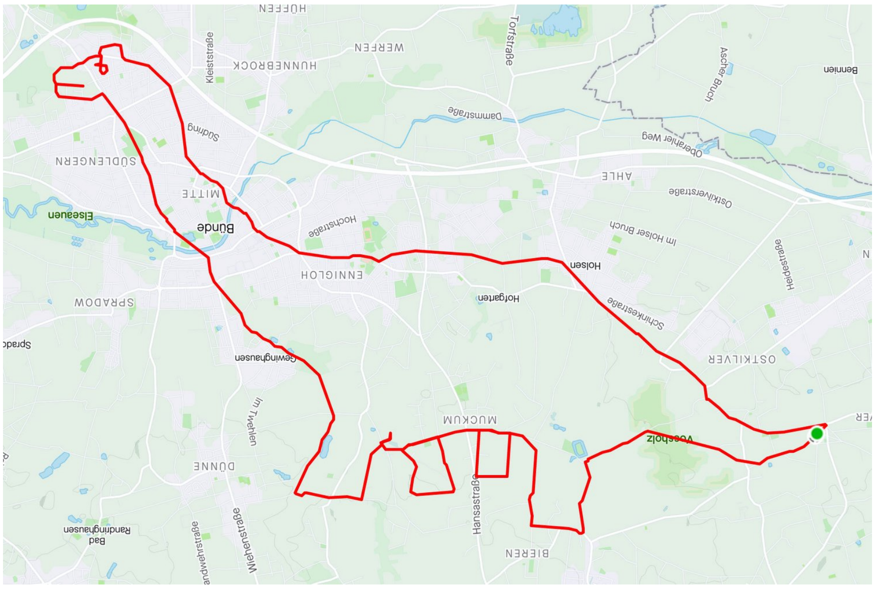


Figure 3.7: A GPS art of a dinosaur, with the map rotated upside down to match the orientation of the presented object. The inverted orientation of the map can be recognized by the text labels which are upside down. Source: link

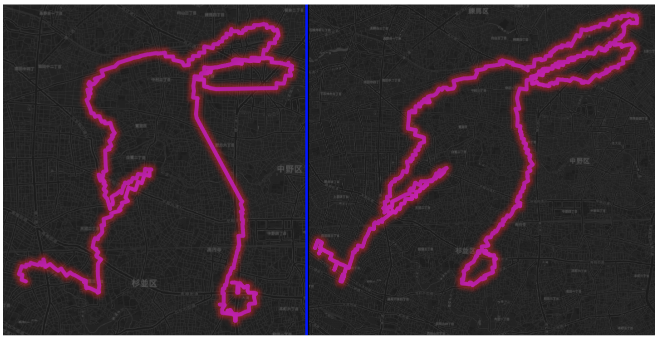


Figure 3.8: Comparison of 2 artistic routes for a drawing of a rabbit. The left image shows a route with no rotation/shearing of the input drawing. The right image shows a route calculated from a rotated and sheared drawing.

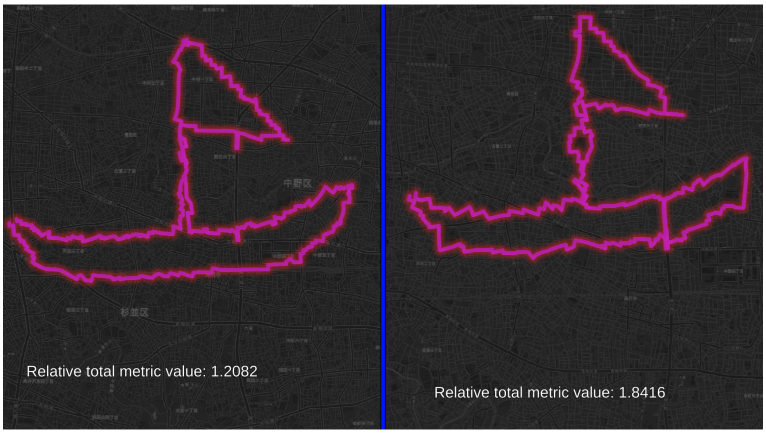

Relative total metric value: 1.2082
Relative total metric value: 1.8416

Figure 3.9: Comparison of 2 artistic routes for a drawing of a sailboat. The left image has a low RTMV, while the right image's value is high.

- length of the edge
- similarity metric between the road (graph) edge and the drawing segment

For the quality evaluation, the total length of the resulting GPS art route has no relevance. Therefore, the similarity metric should be the only thing to be considered. The sum of all the metric values for every edge in the resulting route is referred to as the *Total Metric Value*. This is a relevant numeric measure of how similar the resulting route is to the input drawing based on the similarity metric used in the cost function. To account for the scale factor and enable scale-invariant comparisons, the metric is divided by the total route length to obtain the Relative Total Metric Value (RTMV). Figure 3.9 displays the RTMV in practice. It shows two resulting artistic routes chosen from many candidates. One of them has one of the best metric values, while the other one has one of the worst. A high RTMV means that the generated route has some significant deviations from the source drawing, which in most of the cases also decreases the GPS art quality.

RTMV is one of the components of the evaluation framework. It cannot be used as the only method for assessing artistic route quality. While low RTMV usually means that the route is a good fit to the input drawing, it does not mean that the candidate route with the lowest RTMV is always objectively the best. Considering the aesthetic aspect of GPS art, there are some cases when the best route candidate may not necessarily have the lowest RTMV.

Examples include:

- the route has a significant shear which decreases its perceptual clarity, despite a low RTMV (see figure 3.8).
- the deviation from the input drawing does not decrease its perceptual clarity. For example, a route deviation in the GPS art of an animal can make its tail longer. While the RTMV is increased because of this, the deviation itself does not negatively impact the perceptual clarity.
- there are some outlier segments, but the route is a good fit in general. The outlier segments can be caused by the local layout of the road network. A single outlier

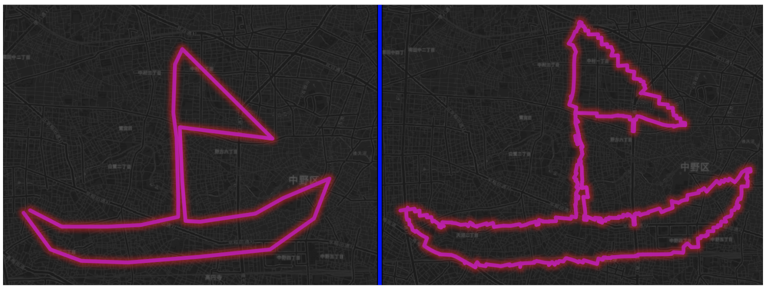


Figure 3.10: Input sailboat drawing and the resulting GPS art route. The result is a good match in general, despite having 2 outlier segments that increase its RTMV

segment introduces a significant increase in the RTMV but by itself it does not visually disqualify the route candidate (see figure 3.10).

Because of the mentioned cases, additional learning-based methods for route quality evaluation are also used (explained in Section 3.3.2).

### 3.3.2 Learning-based evaluation

#### Object classifier metric

As mentioned before, using a machine learning based object classifier is one of the options for quality assessment. Since designing and training a convolutional neural network is out of scope of this thesis, we had to find a model that would demonstrate its applicability to the GPS art case. The idea is to rasterize (convert to image) a candidate artistic route and see how the classifier recognizes it. The resulting class labels and their probabilities can serve as an indicator of the art's quality.

To make this work, the classifier has to have been trained on human-made contour-based drawings. Google *"Quick, Draw!"*[5] is a research project tailored specifically to this purpose. They set up a web application to collect doodle sketches drawn by people, with an explicitly stated goal: to later use it in neural network training. The users of the app were tasked with sketching their own representations of different kinds of objects. The crowdsourced data[6] was made publicly available, which meant that a database of around 50 million hand drawn sketches with 345 different labels could be used by anyone to create a powerful doodle sketch classifier.

Such a dataset is invaluable for research, because every drawing is unique and includes man-made imperfections, as well as different ideas that people have for representations of certain objects. Figure 3.11 shows how much the artists' imagination and skill expression can vary. GPS art is similar in that sense, as it is also often imperfect and meant to express its creator's idea of the object, regardless of how distant it may be from a commonly understood representation.

[5] https://quickdraw.withgoogle.com/

[6] Google *"Quick, Draw!"* data: https://quickdraw.withgoogle.com/data

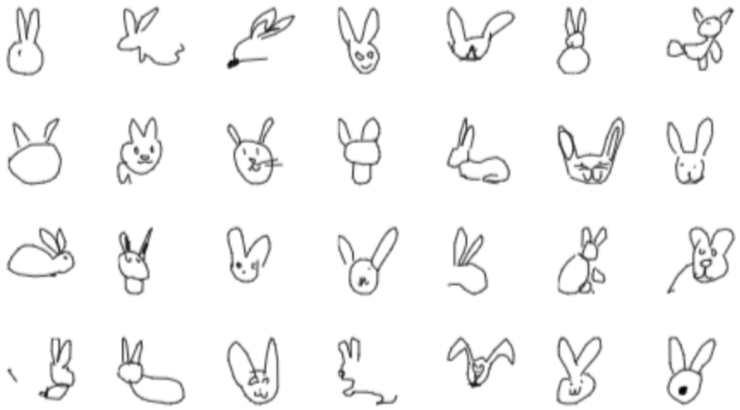


Figure 3.11: Some of the drawings created by users when tasked with drawing a rabbit

Previous investigations by Lars Wachter[7] describe the use of the Google "*Quick, Draw!*" data to train a deep learning based doodle sketch classifier and provide code[8] to reproduce the results. The provided model is used in the evaluation framework and is responsible for labelling the artistic route candidates. An example image classification result is shown in Figure 3.12.

One limitation of using a machine learning method for any task is that it is limited by its training data. The Google "*Quick, Draw!*" dataset could not possibly include all common drawing representations of all objects. In our case, some artistic route candidates could not be recognized properly despite being quite clear.

Other limitations are specific to the GPS art case. One incompatibility with the training dataset is caused by the fact that GPS art drawings are supposed to be single stroke, since we want the route's start and end to be connected by a continuous route. On the other hand, the training dataset does not have such a limitation, which means that the training drawings can be more complex when made up of multiple disconnected components. Another issue is that GPS art drawings can never be perfectly smooth. In real-world scenarios, the underlying road network will only allow for irregular, jagged shape of the drawing. For these reasons, there is a slight disparity between the training dataset and the images that it is meant to label. Perfecting a doodle sketch recognition model is a separate research challenge out of scope of this thesis. An additional challenge would be finding a GPS art dataset of sufficient volume and quality to be used for neural network training.

The usage of the machine learning model for recognition complements the evaluation framework in cases when the RTMV is not sufficient as a metric of quality. For example, if the candidate route has been sheared, the visual impact of such a transformation needs to be assessed. The RTMV value is transformation invariant, meaning that a drawing that is significantly deformed by a shear transformation can still have a good value of this indicator. Using the trained object classifier, a lower label certainty is returned for highly deformed objects.

[7] Lars Wachter's blog: `https://larswaechter.dev/blog/recognizing-hand-drawn-doodles/`

[8] Github repo: `https://github.com/larswaechter/quickdraw-cnn`

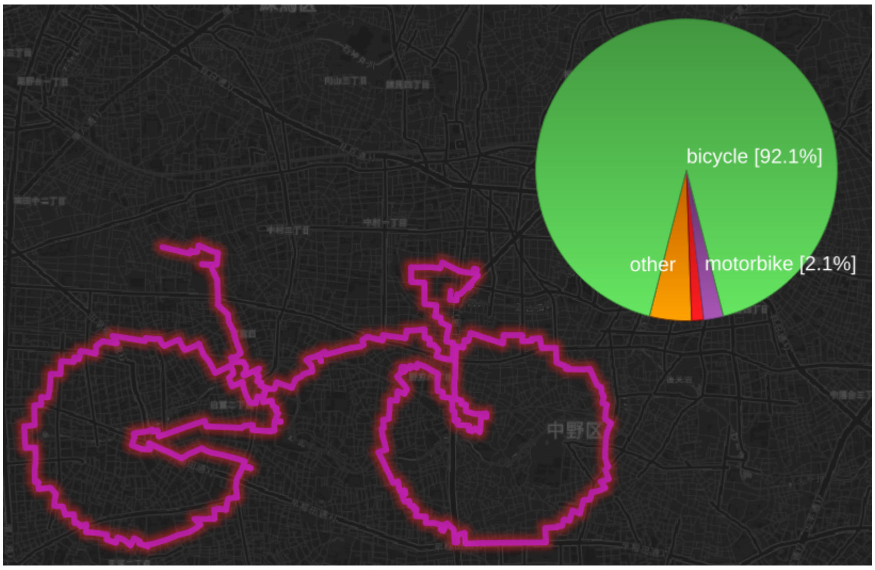


Figure 3.12: Artistic route of a bike displayed next to its classification results. In this case the object classifier returned a 92.1% certainty on the match.

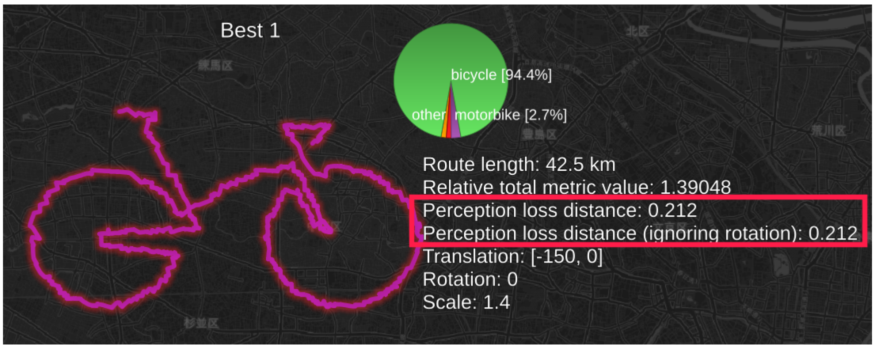
### Perception loss metric

One thing that neither of the previously mentioned evaluation methods covers is the perceptual difference between the input drawing and the output route. Assessing the similarity between two images is a trivial task for humans, but the underlying processes are known to be quite complex in terms of a software implementation. The scientific research by Zhang et al. [2018] explores the use of deep features as a metric of perceptual loss between two images. Their work focuses on recreating something resembling human perception. A summary of their results in their project's website[9] states that their deep feature approach outperforms most of the commonly used perception loss algorithms to date.

In our case, we used the Learned Perceptual Image Patch Similarity (LPIPS) Python library from the project's Github[10] repository. The code allows computing a distance between any pair of images. In our case, those are the image representations of the input drawing and the result artistic route. Since our algorithm outputs the route as a graph, it has to be converted to an image format first. Once that is done, the LPIPS loss function can be used to obtain the distance between the two images. This metric of perception loss allows for a comparison of multiple candidate GPS art routes for the input drawing, to select the optimal choice.

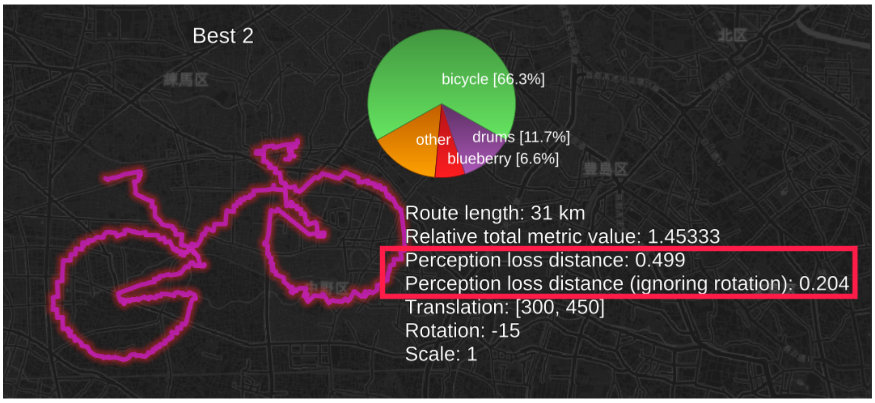
In the visualizations shown in this thesis, we label this metric as *Perception Loss Distance*. There are two versions of this metric, as shown in Figure 3.14. One of them uses images with preserved rotation relative to the input drawing. This is to measure the visual impact introduced by rotation relative to the input shape. The second metric (with the "ignoring rotation" label) measures the rotation-invariant perception loss distance. Figure 3.13 shows the example metric outputs for a route variant with no rotation relative to the input drawing. In this case both the perception distances are equal. Figure 3.14 shows a different case, where

[9] https://richzhang.github.io/PerceptualSimilarity/
[10] Github: https://github.com/richzhang/PerceptualSimilarity



Route length: 42.5 km
Relative total metric value: 1.39048
Perception loss distance: 0.212
Perception loss distance (ignoring rotation): 0.212
Translation: [-150, 0]
Rotation: 0
Scale: 1.4

Figure 3.13: Example evaluation result for an artistic route candidate with no rotation relative to the source drawing



Route length: 31 km
Relative total metric value: 1.45333
Perception loss distance: 0.499
Perception loss distance (ignoring rotation): 0.204
Translation: [300, 450]
Rotation: -15
Scale: 1

Figure 3.14: Example evaluation result for an artistic route candidate with a rotation of 15 degrees counter-clockwise relative to the source drawing

the GPS art has slight relative rotation. This has a major impact on the perceptual similarity metric, as we see that the rotation itself causes an increase from 0.204 to 0.499 distance as computed by the LPIPS library.

The total evaluation score uses only the rotation-invariant perception loss metric, since it is known that the final visualization of any GPS art can be displayed on a map with arbitrary orientation.

## 3.4 Automatic GPS art workflow

### Area of interest

Choosing an area of interest is the simplest constraint and it is done at the preprocessing stage. Once a spatial extent is defined, the road network data can be clipped and the input drawing can be positioned anywhere within it. The transformation parameters in the exhaustive search should ensure that none of the transformed input drawings fall outside of the area of interest. The area should be large enough to contain sufficient street segments

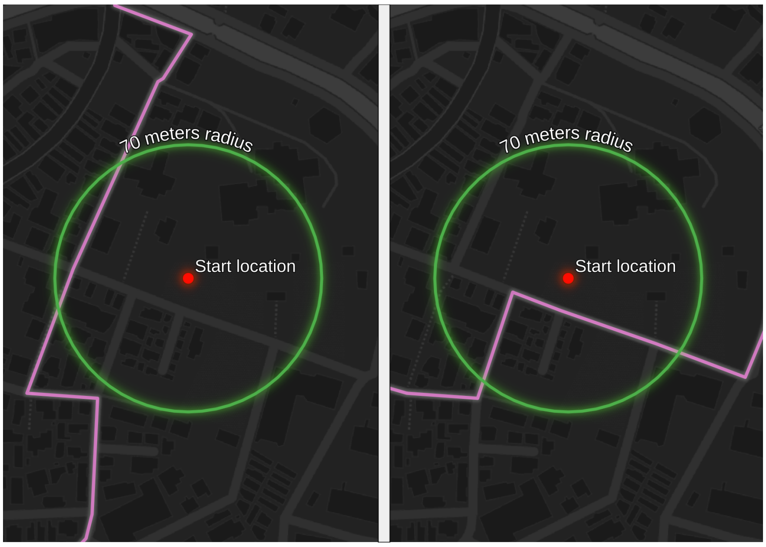


Figure 3.15: Two distinct routes which satisfy the constraint of a starting point (red marker) within a 70-meter threshold

for creating the desired artistic route. This parameter acts as a filter for the routing data but does not introduce any constraints on the route itself.

### Starting point

If a specific starting point, specified by geographic coordinates, is desired, there are two ways to handle it. The first is to use a closed input drawing by connecting the first and last vertex of the drawing segments. When the shape is closed, any of its points can be used as the starting point, and the resulting route will still end up as a single closed component.

The second method uses an unclosed input drawing as-is. This limits the starting point options to only the first or last vertex of the drawing, as any other choice would result in a route that covers only a part of the drawing rather than the entire shape.

To implement the starting point as a soft constraint we allow an additional threshold parameter. All the road network graph nodes which are within the specified distance of the user-selected starting point become the potential starting nodes for the artistic route. The exhaustive search is therefore limited to the routes that pass through one of those nodes, in this context referred to as the starting node of the route. Figure 3.15 shows an example of two different routes which satisfy the starting point requirement with a 70-meter distance threshold.

The route search starts by first applying transformations other than translation. Then, translation is used to anchor the input shape onto the starting node. This operation is performed for each of the shape's points and for each of the candidate starting nodes. The routing algorithm runs for each of these variations, generating *NxM* new route candidates that are guaranteed to visit one of the starting nodes for the *N* points that make up the input shape and *M* graph nodes that are within a specified distance of the user-selected point. This happens for each exhaustive search transformation, so overall there are *NxMxT* candidates,

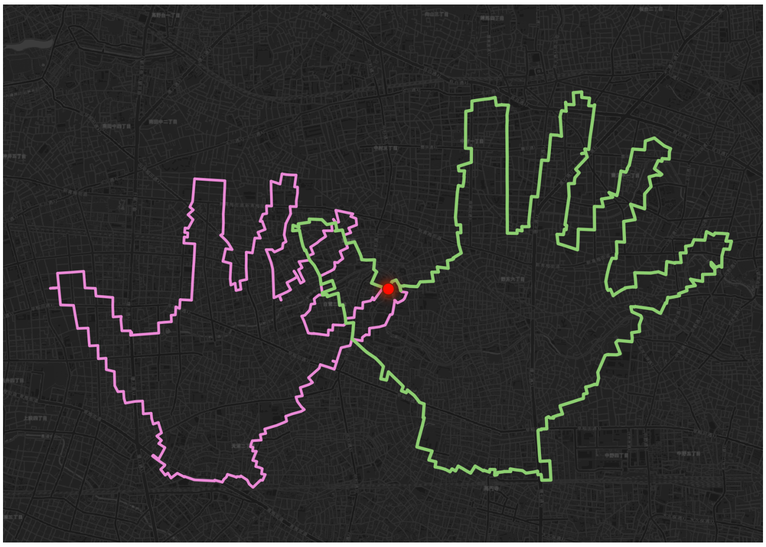


Figure 3.16: Example two distinct artistic route candidates which satisfy the constraint of a starting point (red marker)

where $T$ denotes the number of non-translation transformations in the exhaustive search. Figure 3.16 illustrates two example routes that satisfy the starting point constraint. If there are no streets within the specified distance, then the algorithm uses the street node which is the closest to the selected starting point. This is shown in Figure 3.17, where the resulting route goes through the closest point in the street network, which is at the same time more than 70 meters away from the selected starting location.

### Desired route length

Route length acts as a soft constraint because it is not always possible to guarantee an exact artistic route length. Enforcing a hard constraint for this would limit the number of candidate routes to very few choices, making it difficult to find within a reasonable time. Instead, a certain difference between the desired and obtained route length is allowed. All the generated candidate routes have a length that does not deviate from the desired length by more than a defined threshold value.

To handle the route length requirement, the exhaustive search is enhanced with a slight modification. The first step is to determine the proper scale of the input drawing. It is known that for a given transformation-fixed input drawing, the generated routes can have different lengths depending on the layout of the underlying road network. Therefore, it is not possible to determine a single size of the input shape that will result in routes of consistent length. Instead, we look for a certain scale that is expected to return routing results with the route length close to the required value. This is achieved by first generating the artistic route with an arbitrary initial scale of the drawing. If the route length is greater than required, the input is scaled down; otherwise, it is scaled up. Then the route is generated again. This process is repeated until the output route length is within the specified threshold of the required length. If the threshold value is set too low, the result may never converge if it does not find

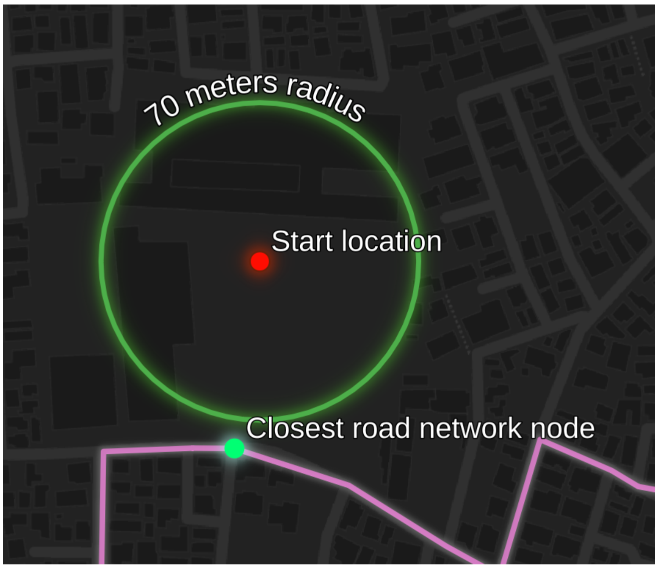


Figure 3.17: A starting point (red marker) and the resulting route (pink) in a case when the chosen starting point is further than 70 meters from any roads.

an appropriate scale for the shape. Table 3.1 demonstrates the number of iterations needed to find a proper scale for 42 kilometer routes with a 500 meter threshold value.

Once the scale value has been determined, the exhaustive search can proceed with affine transformations and generating the artistic route candidates. The scaling factor is then fixed for all the applied transformations, with the expectation that the lengths of the generated routes will be close to the required value. This is not always the case, because of the natural irregularity of the road network. Figure 3.18 illustrates one such case, where two routes generated for equally scaled input in the same area of interest have significant differences in length. The inputs have different rotation and placement, which is enough to introduce a length disparity in the resulting routes. If the user requires extra precision in the route length, the previously described length refinement method can be reused on the generated candidate routes. This will ensure that the route length stays within a certain threshold for all the candidates, at a cost of lower efficiency because of more routing tasks needed to refine each route's length.

As a final consideration, the evaluation framework can be extended to enable better route selection based on specific length requirements. When evaluating potential artistic routes, the proximity to the desired length is added as a component of the overall score. This means that the score will slightly favor routes that are closer to the desired length, while still taking into account the visual quality of the route.

### Generalization of the input drawing

In the GPS art workflow, the routing component (which is explained in Section 3.1.1) is responsible for identifying a path for each segment of the input shape. However, in certain situations, such as when a drawing segment is extremely short, the routing algorithm may select the same road network node as the starting and ending point of the route if that node

| **Test input** | **Iterations to find proper scale** |
|---|---|
| Bike | 14 |
| Dolphin | 4 |
| Elephant | 7 |
| Hand | 20 |
| Lightning | 6 |
| Rabbit | 7 |
| Sailboat | 27 |
| **Average** | **12.1** |

Table 3.1: Number of iterations needed to find a scale of the drawing which gives a desired route length of 42 kilometers (+- 500 meters) in the Tokyo road network.

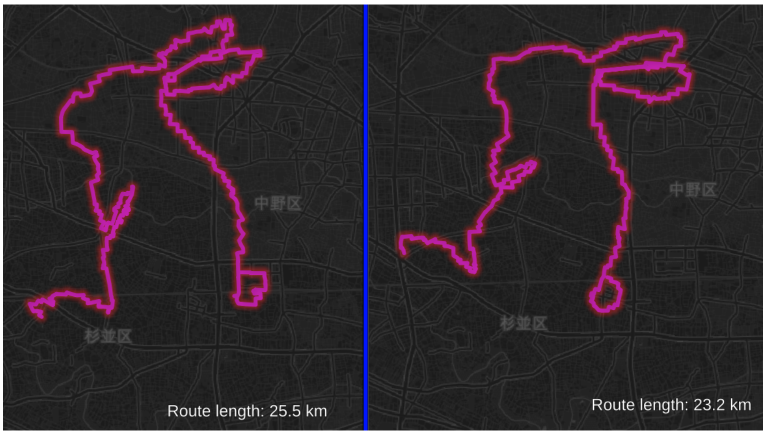
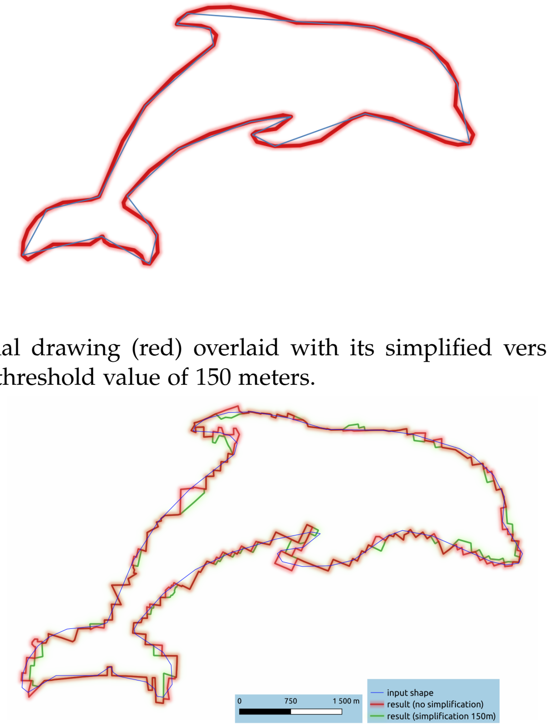


Figure 3.18: Two artistic routes generated from an input drawing with a constant scale. The route to the left is 25.5 kilometers long. The route to the right is 23.2 kilometers long.



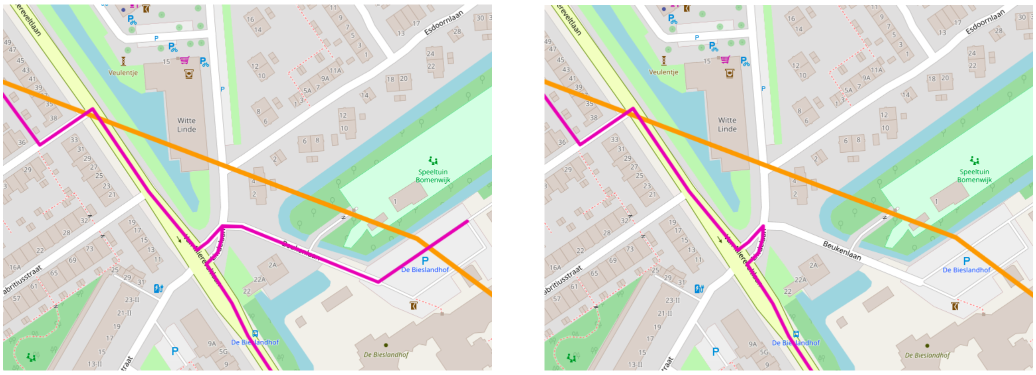
(a) Original drawing (red) overlaid with its simplified version (blue) with a threshold value of 150 meters.

(b) GPS art results. Red - without simplification of the input, green - simplification threshold of 150 meters. Thin blue line is the original, non-simplified input.

Figure 3.19: Figures demonstrating the impact of input shape simplification.

happens to be the closest option to both ends of the drawing segment. Furthermore, when the granularity of the drawing is high, the GPS art workflow may experience reduced efficiency due to the large number of routing tasks required for each small segment of the input shape. To address this issue, an additional generalization step can be introduced during the preprocessing stage. This involves reducing the resolution of the input shape by simplifying its geometry to remove any segments that are shorter than a certain threshold value. The simplification is done using Douglas-Pecker algorithm[11]. Selecting an appropriate threshold value ensures that the generalized drawing retains its visual meaning as much as possible, while avoiding the inclusion of any redundant data in the routing algorithm. Figure 3.19a shows an example of an input drawing that was oversimplified and lost too much of its visual quality. Figure 3.19b shows the impact of simplification on the resulting artistic routes. It can be observed that the route generated with a simplified input deviates more from the original non-simplified shape. The route itself does not become more simplified, because it simply follows the segments of a simplified shape using the underlying road network. The purpose of simplification in this case is not to achieve a simpler result, but to make sure that the input is as simple as possible for the sake of efficiency of the GPS art algorithm.

[11] https://en.wikipedia.org/wiki/Ramer%E2%80%93Douglas%E2%80%93Peucker_algorithm



(a) Before removing U-turns

(b) After removing U-turns

Figure 3.20: Example of a routing scenario when a U-turn occurs. Input drawing segments are colored orange. Output artistic route is colored pink. A topographic map is shown in the background.

### Removing U-turns on dead ends

The GPS art algorithm introduced by Waschk and Krüger [2018] served as a solid foundation for the GPS art workflow presented in this thesis. Any route generated from an input single stroke drawing is made up of a single connected component. One point of improvement is the handling of frequently occurring U-turns, which significantly impact the visual clarity of artistic routes. A U-turn is created between two routes that come from two subsequent segments when the first route shares more than one node with the second one. This case is illustrated in Figure 3.20a. The full artistic route is still valid, but the U-turn introduces some undesired effects. It creates a visual dead-end segment that decreases the quality of the GPS art. Furthermore, it is inconvenient for the person traveling along the route.

To address this issue, an extra cleaning step is added after generating an artistic route. The final route is stored as a list of node pairs (segments), and we identify all the repeated pairs. Repetitions are counted regardless of the pair order, meaning that segments $\overline{AB}$ and $\overline{BA}$ are considered the same. All repeated pairs are removed, thus removing the route parts that look like dead ends (see Figure 3.20b). Because of this change, the route is no longer guaranteed to visit the nodes closest to the ends of the drawing segments. Additionally, removing some route segments after routing itself means that the overall route cost is changed. The RTMV has to be updated accordingly for the purpose of score evaluation. Another consequence is that the route length will decrease because of the removed segments. Since this step can be applied only after the routing is done, it may happen that the route will no longer meet the length requirements and the process would have to be restarted with different parameters.

### GPX export

GPX is a format used to describe routes supported by most physical activity tracking applications. In our experiments we mostly used WKT as a format for outputting and displaying the GPS art routes. To allow easier integration with mobile apps (e.g., Strava, Runkeeper) the final route is converted from WKT to GPX using *ogr2ogr* from the GDAL software suite.

## 3.5 Interactive web application

To consider interactive inputs from the user, a GPS art web application with a graphical interface has been developed in the scope of this thesis. It uses some of the solutions developed for the automatic GPS art workflow (Section 2.1.4). The template matching step is skipped, because a user can visually assess the road network layout to decide where to put the shape. The user is given a possibility to transform (scale, rotate, translate) the shape and get instant feedback in the form of the displayed artistic route, generated in real time after each change to the input drawing is applied. Figure 3.21 shows the state of the app after the transformation is applied. An overlay of the generated artistic route is visualized on top of the input. The control panel (see ”Layer controls” in Figure 3.22) gives the possibility to disable visibility of the input shape and have an unbiased look at the artistic route. The user can also refine the result using exhaustive search and some extra requirements like a starting point or desired route length. Results of such a query are visible on the map in Figure 3.22. The application displays the best route that it could find to match the requirements of the starting point (red dot with blue outline) and the length of 21 kilometers. For the user interactive approach, the exhaustive search transformation parameters are tuned to only search close to the location where the drawing was placed on the map. The aim is to return the best artistic route in the vicinity, considering the user's requirements. For efficiency reasons, the candidate routes are not evaluated using the evaluation framework. Instead, the scoring uses a method based only on the metric value (RTMV) and how close the result route length is to the desired length. This results in a quicker response time and consequently more satisfaction from using the app. The user can inspect the visual quality of the results and choose to continue modifying the route parameters or the input shape properties. Once all the processing is done and the results approved, the route can be downloaded as a GPX file (see ”Export” button in Figure 3.22).

An advantage of this solution, is that it gives the opportunity for a smart choice of the drawing location taking into account the distinct characteristics of the area. The user can inspect the topographic map and immediately see which locations have a higher potential than others. For example, areas with big farm fields and a sparse road network can be avoided. As a comparison, the automatic GPS art workflow (Section 2.1.4) does not have any external knowledge, apart from the road network data, which means that it will exclude bad results only after obtaining them.

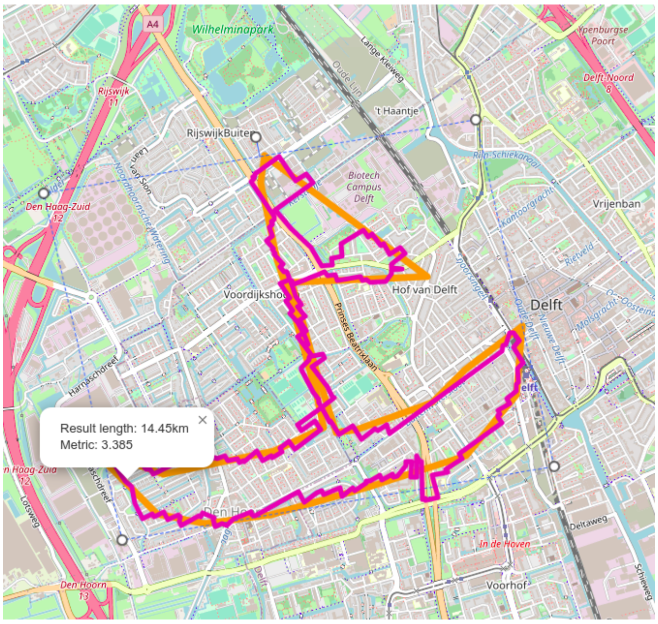
One limitation of this solution is that it is not fully automatic and requires user input. Because of the added manual work, the number of route options explored will usually be lower than in the case of the automatic GPS art workflow.

## 3.6 Datasets and tools used

### 3.6.1 Data sources

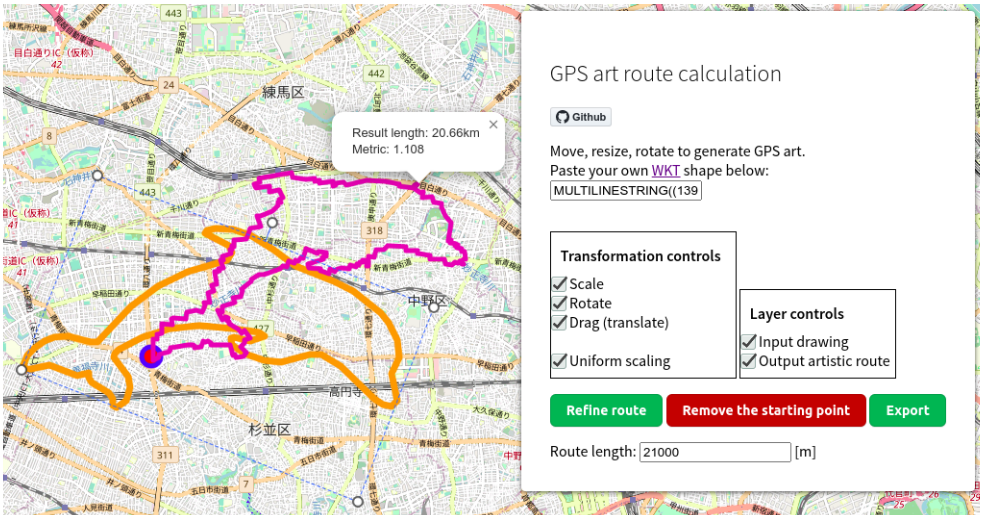
Having road network data from various locations was crucial for the development and tests. Two open road network datasets were used in the end:

- OpenStreetMap (Wiki [2022])



Result length: 14.45km
Metric: 3.385

Figure 3.21: View of the GPS art application after a transformation to the input shape (orange) is applied. The application instantly displays the artistic route (pink) generated for the current form of the input.



Result length: 20.66km
Metric: 1.108

GPS art route calculation

Github

Move, resize, rotate to generate GPS art.
Paste your own WKT shape below:
MULTILINESTRING((139

**Transformation controls**

☑ Scale
☑ Rotate
☑ Drag (translate)

☑ Uniform scaling

**Layer controls**

☑ Input drawing
☑ Output artistic route

Refine route | Remove the starting point | Export

Route length: 21000 [m]

Figure 3.22: Graphical user interface of the GPS art app. The control panel is visible on the right. The red dot with a blue outline marks the starting point. The input drawing is colored orange and the output artistic route is colored pink. Upon hovering over the route, a text is displayed with the route's length and its metric value.

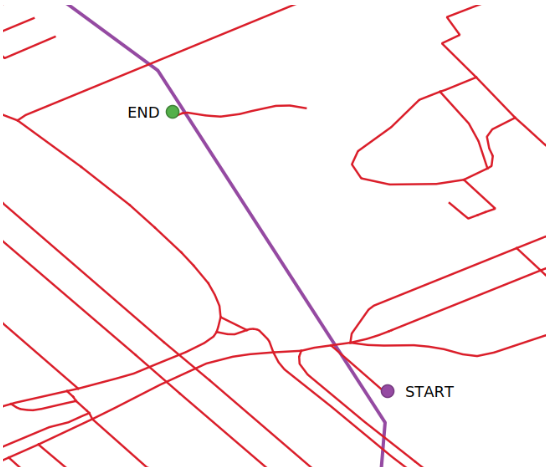


Figure 3.23: Example of a disconnected street segment (red) which causes a routing failure. The routing start and end node are labelled. The input shape segments are colored purple.

- Urban Road Network Data (Networks [2016])

*OpenStreetMap* was slightly more challenging to use, since it is not strictly a road network dataset, but rather a comprehensive topographic one. Additional data cleaning was required, for example to get rid of disconnected components in the road network graph to be used for routing. Figure 3.23 illustrates an example that causes the routing algorithm to fail. Fixing these cases was made possible by the *pgr_connectedComponents* function from the *PgRouting* software package (see Table 3.3). This function enables finding disconnected graph components, which can then be deleted or merged, depending on the case.

### 3.6.2 Software tools

The presented solutions were implemented using a variety of programming languages and libraries, summarized in table 3.2. Other software tools used in the development phase are shown in table 3.3.

The exhaustive search method, routing and the backend for the interactive application were implemented in C++. *Boost* library provides data types and functions for routing. It also has useful functionality to manipulate geometric types. *CGAL* library serves to construct a k-d tree of the road network nodes. For each routing task (for each segment of the input drawing) the start and end nodes are efficiently selected using a nearest neighbor search from the end points of the drawing segment. *Libpqxx* is used to fetch road network data stored in a *PostgreSQL* database. Finally, *CROW* library is used to build the REST API and the backend server of the interactive application. The C++ functions for generating GPS art are exposed as API endpoints to be consumed by the frontend component of the application.

Template matching, automatic GPS art workflow and evaluation framework were implemented using *Python* and its libraries. *Shapely* library is used for manipulating geometric types, for example by applying affine transformations. *OpenCV* is used for operations on images and template matching. *Pandas* is used for data filtering and processing in the evaluation framework. *Tensorflow* enables object recognition in images with the pre-trained machine learning model. *LPIPS* library provides a perceptual loss function for calculating a visual distance between input images and the images of the output routes. All the developed components were integrated using standard *Python* functionality to complete the automatic GPS art workflow.

The interactive application frontend was developed using *Javascript*. *Leaflet* serves as a base for the interactive map and allowed visualization of input geometries and the resulting routes. A separate *Leaflet* plugin called *L.Path.Drag* is used to enable interactive transformation of the shapes on the map. It provides a useful functionality of dragging, scaling and rotating equipped with an event API that allows real-time route generation after any kind of transformation is applied by the user. Each change of the input shape emits an event followed by a request to the C++ application backend, which returns the route geometry in *WKT* format. After a conversion to another format using *WKX* library, the route can be displayed on the map.

Other software tools were also used in the development and test phase. *QGIS* was useful for visualizations of all the spatial data that appeared at various stages. *ogr2ogr* tool enabled converting geometries in vector format to raster (image) and *GPX* format. A *PostgreSQL* database served as a storage and processing tool for the road network datasets used in tests. Import of *OpenStreetMap* routing data for the test locations was achieved using *Osm2pgrouting*

| Components | Programming language | Library | Used functionality |
|---|---|---|---|
| - Exhaustive search<br>- Interactive application backend | C++ | Boost | Graph data types, geometric types, routing functions |
| | | CGAL | K-d tree type, nearest neighbor search |
| | | libpqxx | Querying road network graph data stored in a PostgreSQL database |
| | | CROW | Rest API and server for the interactive application |
| - Template matching<br>- Automatic GPS art workflow<br>- Evaluation framework | Python | Shapely | Geometric types, basic geometric operations |
| | | OpenCV | Image operations, template matching |
| | | Pandas | Manipulation of data for the evaluation framework |
| | | Tensorflow | Object recognition in images using a pre-trained model |
| | | LPIPS | Perceptual loss function |
| - Interactive application frontend | Javascript | Leaflet | Interactive map, visualization of data on the map |
| | | L.Path.Drag | Leaflet plugin for transforming shapes interactively by dragging, scaling, rotating |
| | | WKX | Conversions between different geometry formats (GeoJSON, WKT) |

Table 3.2: Overview of the programming languages and libraries used in the experiments

| Software package | Usage |
|---|---|
| QGIS | Visualizations of geometric data in the development phase |
| ogr2ogr | Exporting routes to GPX format, converting vector geometries to rasters (images) |
| PostgreSQL | Storage and processing of road network data |
| PgRouting | PostgreSQL extension for processing graph data |
| osm2pgrouting | Import of OSM routing data to a PostgreSQL database |

Table 3.3: Overview of the software packages used in the experiments

# 4 Results and discussion

This chapter presents the research results in a summarised manner. Many illustrations will be used, because they are crucial to conveying most of the results related to generating GPS art. Automatic GPS art workflow and user interactive approach are also compared in terms of their strengths and weaknesses considering various scenarios. In the last section, the performance and innovative aspects of the products of this thesis are discussed.

## 4.1 Test datasets

The automatic GPS art workflow (described in Chapter 3) has been tested with 3 chosen input drawings in different locations around the world, mostly in urban and suburban areas. One location (Delft) contains some rural areas to show what impact they have on artistic routes.

The chosen locations are:

- Tokyo, Japan. Map projection: EPSG:3100
- Paris, France. Map projection: EPSG:32631
- New York City, USA. Map projection: EPSG:32618
- Delft, Netherlands. Map projection: EPSG:28992
- Amsterdam, Netherlands. Map projection: EPSG:28992

The chosen input drawings (Figure 4.1) are:

- bike
- elephant
- hand

## 4.2 Test results

Visual test results are displayed in the figures below (Figure 4.2, Figure 4.3, Figure 4.4, Figure 4.5, Figure 4.6). The full results in tabular format are shown in Table 4.1.

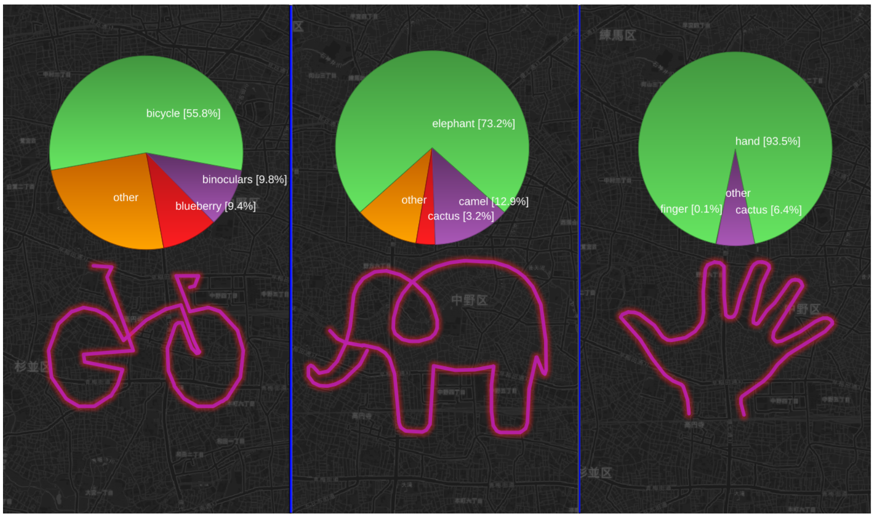


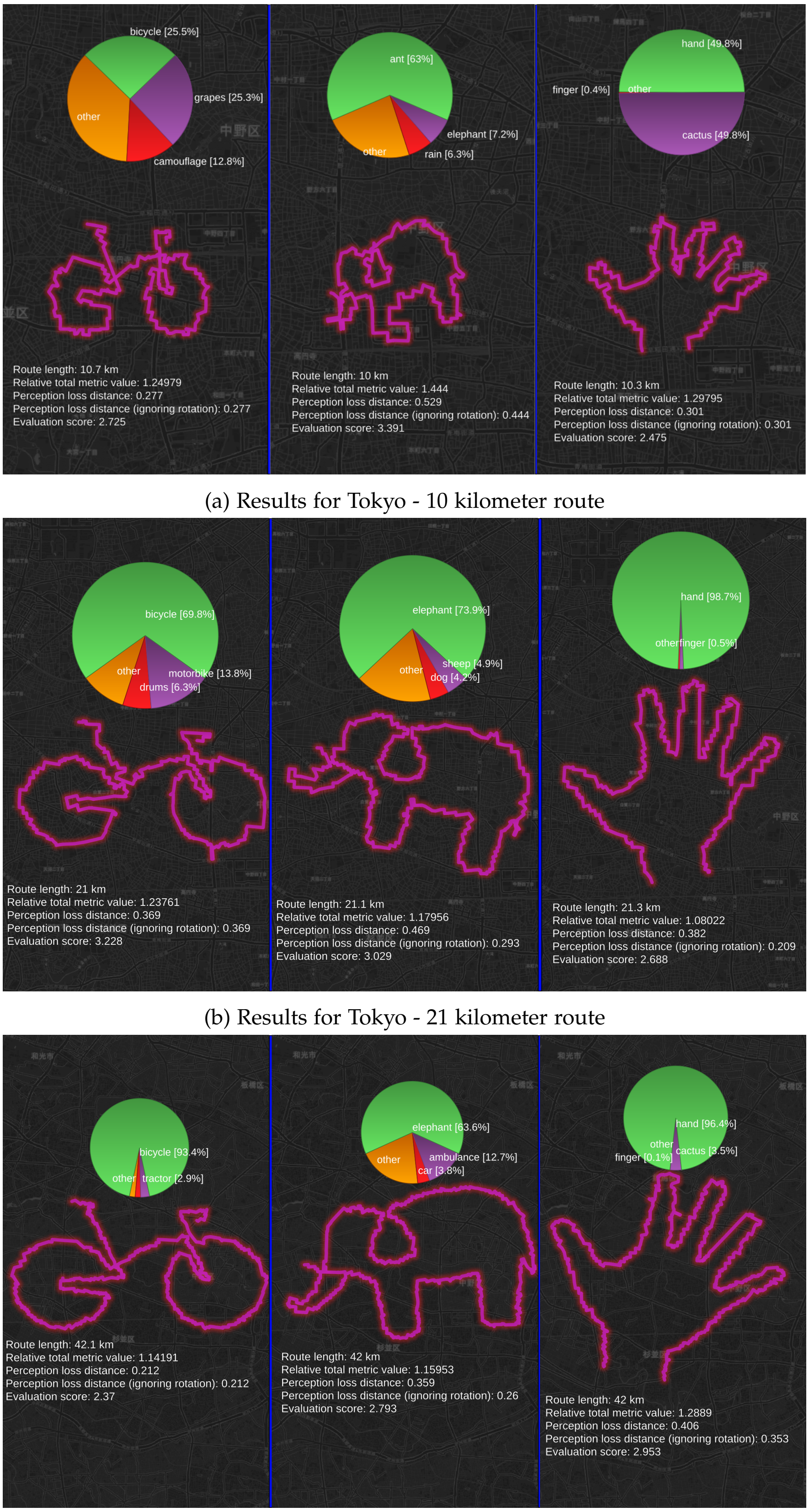
Figure 4.1: Input drawings chosen for the tests. The diagram above them shows reference labelling given by the object classifier.

### 4.2.1 Visual analysis

A general conclusion is that the chances of finding a suitable artistic route are higher as the route length increases. This is expected and has a simple explanation. Assuming that the resolution (density) of the road network is comparable for the entire region, the deviations from the road network are also always comparable. As the GPS art is scaled up, these deviations appear proportionately smaller and thus their negative impact on the art's quality is decreased. This is why longer artistic routes appear to be more smooth than their shorter versions, where the deviations from the source drawing are relatively bigger. A visual example is shown in Figure 4.7 which shows routes for the same shape in 4 varying scales.

Because of the combination of various quality metrics, the algorithm is capable of choosing a route that is not deformed by topographic obstacles. An example of this behaviour is visible in Figure 4.3b in the GPS art of an elephant. The street network in the centre of Paris is split by a wide barrier formed by the river Seine. This creates a substantial area with lower street density and choke points wherever the river is crossed by a bridge. The artistic route has to properly fit the roads that cross the river, otherwise they would introduce too much of a detour that would lower the route quality. The result looks successful, with the route navigating around the river and still representing something that resembles a rotated elephant.

Another observation is based on the visual disparity in the same kinds of objects depending on the city. Differences in the spatial layout of the road networks in the test locations becomes apparent. Tokyo's road network (Figure 4.2) seems to be the most dense out of the test datasets, providing visually acceptable solutions even for the 10-kilometer routes. In the rest of the locations, the 10-kilometer routes are less visually appealing, with significant deviations from the original shape. Results from New York City (Figure 4.4) are the most distinguishable, because of the characteristic square grid street layout typical for the cities of North America. This property results in more horizontal and vertical straight line segments in the artistic routes. For example, the elephant in Figure 4.4b has almost perfectly

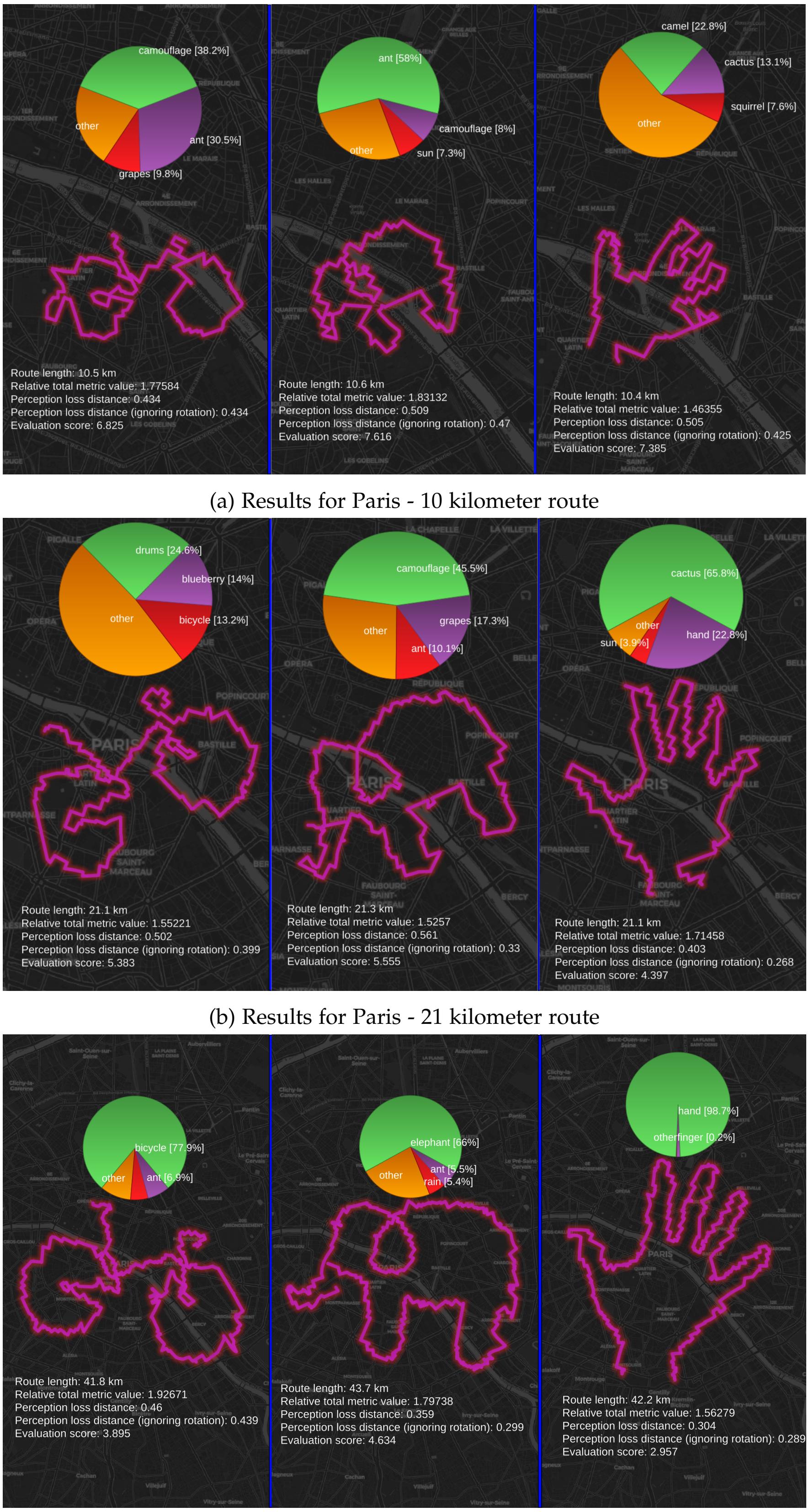

bicycle [25.5%]
grapes [25.3%]
other
camouflage [12.8%]
中野区
Route length: 10.7 km
Relative total metric value: 1.24979
Perception loss distance: 0.277
Perception loss distance (ignoring rotation): 0.277
Evaluation score: 2.725
ant [63%]
elephant [7.2%]
other
rain [6.3%]
Route length: 10 km
Relative total metric value: 1.444
Perception loss distance: 0.529
Perception loss distance (ignoring rotation): 0.444
Evaluation score: 3.391
hand [49.8%]
finger [0.4%]
other
cactus [49.8%]
中野区
Route length: 10.3 km
Relative total metric value: 1.29795
Perception loss distance: 0.301
Perception loss distance (ignoring rotation): 0.301
Evaluation score: 2.475
bicycle [69.8%]
other
motorbike [13.8%]
drums [6.3%]
Route length: 21 km
Relative total metric value: 1.23761
Perception loss distance: 0.369
Perception loss distance (ignoring rotation): 0.369
Evaluation score: 3.228
elephant [73.9%]
other
sheep [4.9%]
dog [4.2%]
Route length: 21.1 km
Relative total metric value: 1.17956
Perception loss distance: 0.469
Perception loss distance (ignoring rotation): 0.293
Evaluation score: 3.029
hand [98.7%]
otherfinger [0.5%]
中野区
Route length: 21.3 km
Relative total metric value: 1.08022
Perception loss distance: 0.382
Perception loss distance (ignoring rotation): 0.209
Evaluation score: 2.688
bicycle [93.4%]
other
tractor [2.9%]
Route length: 42.1 km
Relative total metric value: 1.14191
Perception loss distance: 0.212
Perception loss distance (ignoring rotation): 0.212
Evaluation score: 2.37
elephant [63.6%]
other
ambulance [12.7%]
car [3.8%]
Route length: 42 km
Relative total metric value: 1.15953
Perception loss distance: 0.359
Perception loss distance (ignoring rotation): 0.26
Evaluation score: 2.793
hand [96.4%]
other
finger [0.1%]
cactus [3.5%]
Route length: 42 km
Relative total metric value: 1.2889
Perception loss distance: 0.406
Perception loss distance (ignoring rotation): 0.353
Evaluation score: 2.953

(a) Results for Tokyo - 10 kilometer route

(b) Results for Tokyo - 21 kilometer route

(c) Results for Tokyo - 42 kilometer route

Figure 4.2: Visualized test results for Tokyo

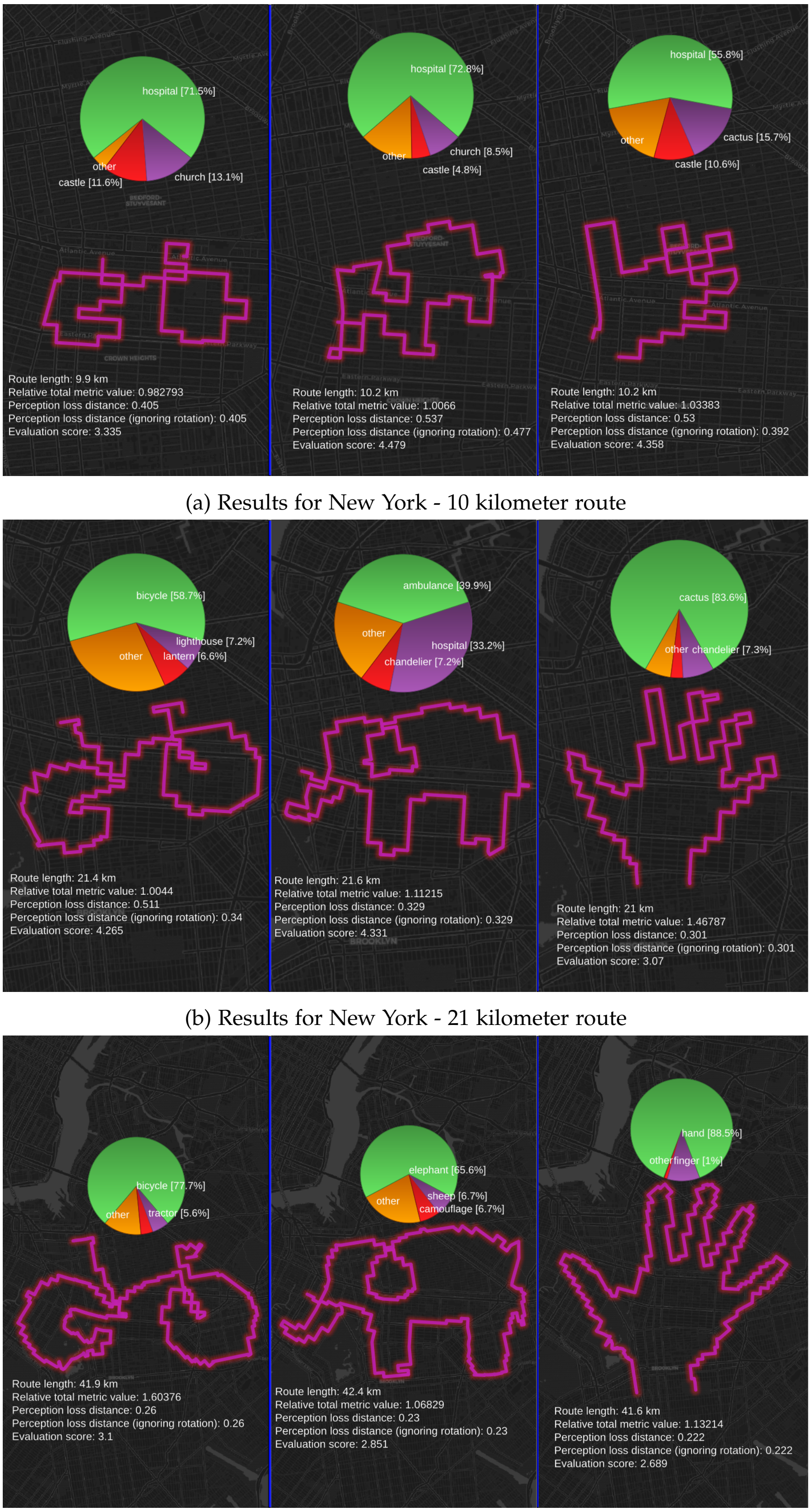

camouflage [38.2%]
other
ant [30.5%]
grapes [9.8%]
Route length: 10.5 km
Relative total metric value: 1.77584
Perception loss distance: 0.434
Perception loss distance (ignoring rotation): 0.434
Evaluation score: 6.825
ant [58%]
other
camouflage [8%]
sun [7.3%]
Route length: 10.6 km
Relative total metric value: 1.83132
Perception loss distance: 0.509
Perception loss distance (ignoring rotation): 0.47
Evaluation score: 7.616
camel [22.8%]
cactus [13.1%]
squirrel [7.6%]
other
Route length: 10.4 km
Relative total metric value: 1.46355
Perception loss distance: 0.505
Perception loss distance (ignoring rotation): 0.425
Evaluation score: 7.385
drums [24.6%]
blueberry [14%]
other
bicycle [13.2%]
Route length: 21.1 km
Relative total metric value: 1.55221
Perception loss distance: 0.502
Perception loss distance (ignoring rotation): 0.399
Evaluation score: 5.383
camouflage [45.5%]
grapes [17.3%]
other
ant [10.1%]
Route length: 21.3 km
Relative total metric value: 1.5257
Perception loss distance: 0.561
Perception loss distance (ignoring rotation): 0.33
Evaluation score: 5.555
cactus [65.8%]
other
sun [3.9%]
hand [22.8%]
Route length: 21.1 km
Relative total metric value: 1.71458
Perception loss distance: 0.403
Perception loss distance (ignoring rotation): 0.268
Evaluation score: 4.397
bicycle [77.9%]
other
ant [6.9%]
Route length: 41.8 km
Relative total metric value: 1.92671
Perception loss distance: 0.46
Perception loss distance (ignoring rotation): 0.439
Evaluation score: 3.895
elephant [66%]
other
ant [5.5%]
rain [5.4%]
Route length: 43.7 km
Relative total metric value: 1.79738
Perception loss distance: 0.359
Perception loss distance (ignoring rotation): 0.299
Evaluation score: 4.634
hand [98.7%]
otherfinger [0.2%]
Route length: 42.2 km
Relative total metric value: 1.56279
Perception loss distance: 0.304
Perception loss distance (ignoring rotation): 0.289
Evaluation score: 2.957

(a) Results for Paris - 10 kilometer route

(b) Results for Paris - 21 kilometer route

(c) Results for Paris - 42 kilometer route

Figure 4.3: Visualized test results for Paris



Route length: 9.9 km
Relative total metric value: 0.982793
Perception loss distance: 0.405
Perception loss distance (ignoring rotation): 0.405
Evaluation score: 3.335

Route length: 10.2 km
Relative total metric value: 1.0066
Perception loss distance: 0.537
Perception loss distance (ignoring rotation): 0.477
Evaluation score: 4.479

Route length: 10.2 km
Relative total metric value: 1.03383
Perception loss distance: 0.53
Perception loss distance (ignoring rotation): 0.392
Evaluation score: 4.358

(a) Results for New York - 10 kilometer route

Route length: 21.4 km
Relative total metric value: 1.0044
Perception loss distance: 0.511
Perception loss distance (ignoring rotation): 0.34
Evaluation score: 4.265

Route length: 21.6 km
Relative total metric value: 1.11215
Perception loss distance: 0.329
Perception loss distance (ignoring rotation): 0.329
Evaluation score: 4.331

Route length: 21 km
Relative total metric value: 1.46787
Perception loss distance: 0.301
Perception loss distance (ignoring rotation): 0.301
Evaluation score: 3.07

(b) Results for New York - 21 kilometer route

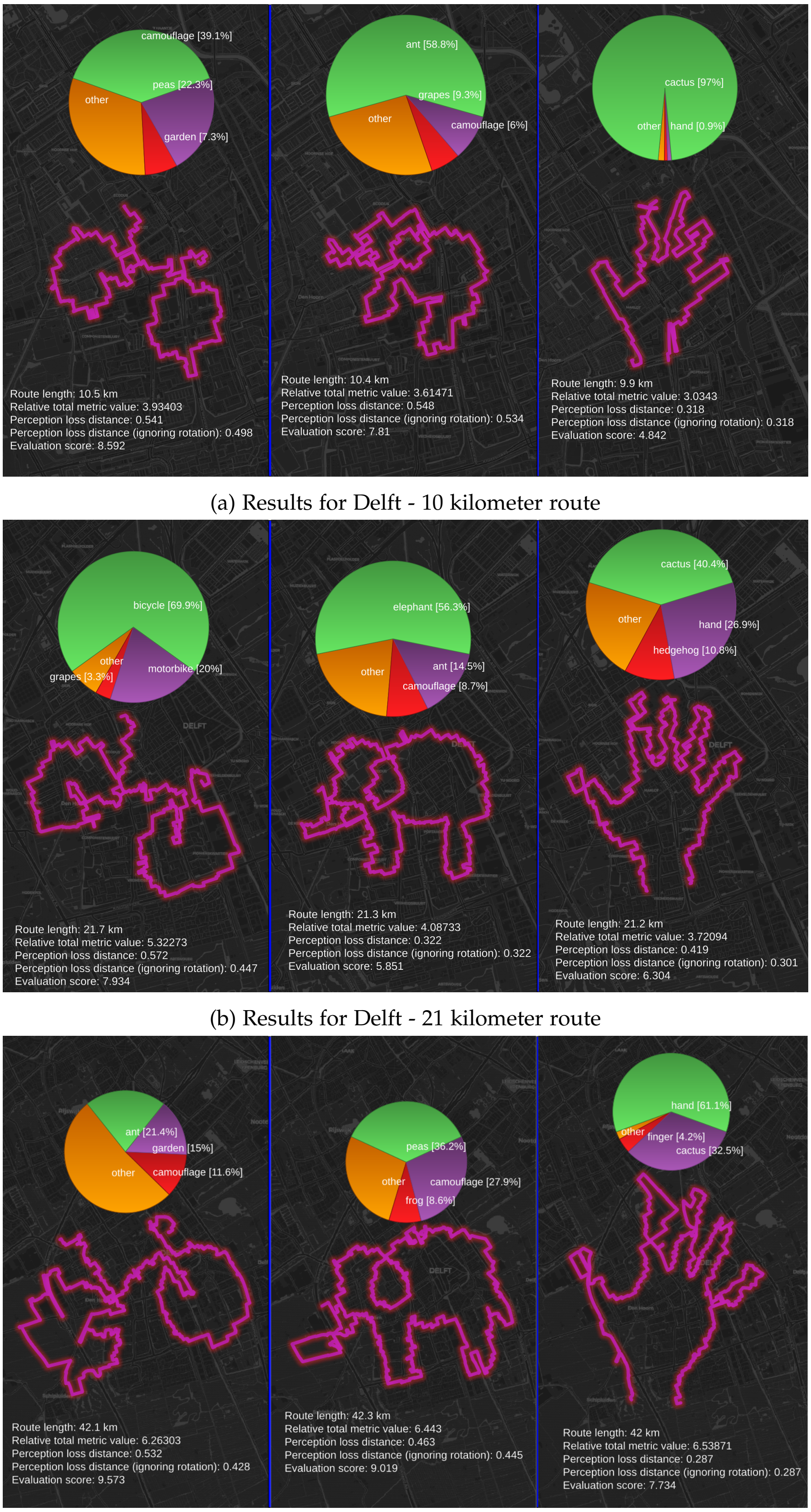
Route length: 41.9 km
Relative total metric value: 1.60376
Perception loss distance: 0.26
Perception loss distance (ignoring rotation): 0.26
Evaluation score: 3.1

Route length: 42.4 km
Relative total metric value: 1.06829
Perception loss distance: 0.23
Perception loss distance (ignoring rotation): 0.23
Evaluation score: 2.851

Route length: 41.6 km
Relative total metric value: 1.13214
Perception loss distance: 0.222
Perception loss distance (ignoring rotation): 0.222
Evaluation score: 2.689

(c) Results for New York - 42 kilometer route

Figure 4.4: Visualized test results for New York

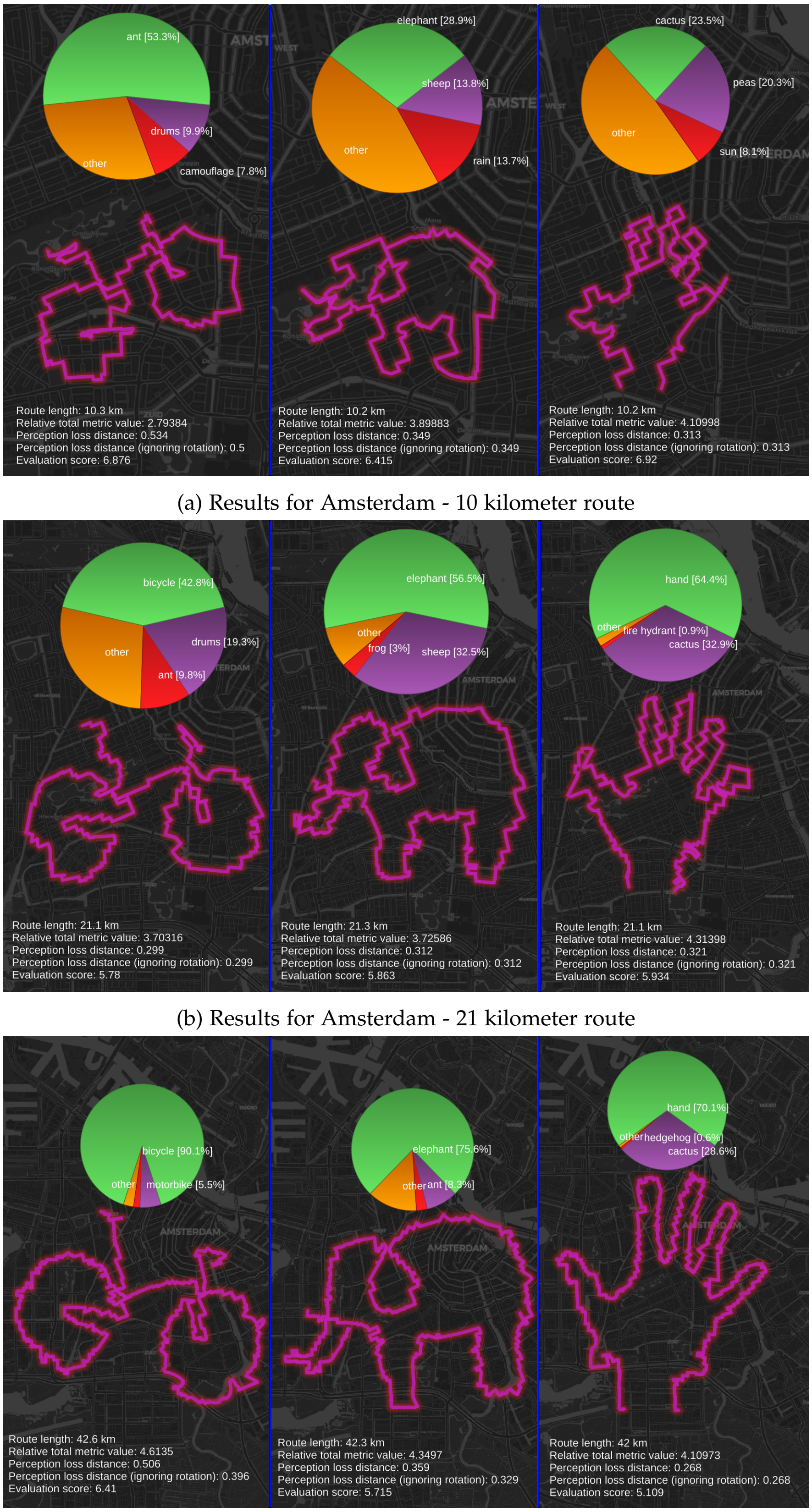

camouflage [39.1%]
peas [22.3%]
other
garden [7.3%]
Route length: 10.5 km
Relative total metric value: 3.93403
Perception loss distance: 0.541
Perception loss distance (ignoring rotation): 0.498
Evaluation score: 8.592
ant [58.8%]
grapes [9.3%]
other
camouflage [6%]
Route length: 10.4 km
Relative total metric value: 3.61471
Perception loss distance: 0.548
Perception loss distance (ignoring rotation): 0.534
Evaluation score: 7.81
cactus [97%]
other
hand [0.9%]
Route length: 9.9 km
Relative total metric value: 3.0343
Perception loss distance: 0.318
Perception loss distance (ignoring rotation): 0.318
Evaluation score: 4.842
bicycle [69.9%]
other
grapes [3.3%]
motorbike [20%]
Route length: 21.7 km
Relative total metric value: 5.32273
Perception loss distance: 0.572
Perception loss distance (ignoring rotation): 0.447
Evaluation score: 7.934
elephant [56.3%]
other
ant [14.5%]
camouflage [8.7%]
Route length: 21.3 km
Relative total metric value: 4.08733
Perception loss distance: 0.322
Perception loss distance (ignoring rotation): 0.322
Evaluation score: 5.851
cactus [40.4%]
other
hand [26.9%]
hedgehog [10.8%]
Route length: 21.2 km
Relative total metric value: 3.72094
Perception loss distance: 0.419
Perception loss distance (ignoring rotation): 0.301
Evaluation score: 6.304
ant [21.4%]
garden [15%]
other
camouflage [11.6%]
Route length: 42.1 km
Relative total metric value: 6.26303
Perception loss distance: 0.532
Perception loss distance (ignoring rotation): 0.428
Evaluation score: 9.573
peas [36.2%]
other
camouflage [27.9%]
frog [8.6%]
Route length: 42.3 km
Relative total metric value: 6.443
Perception loss distance: 0.463
Perception loss distance (ignoring rotation): 0.445
Evaluation score: 9.019
hand [61.1%]
other
finger [4.2%]
cactus [32.5%]
Route length: 42 km
Relative total metric value: 6.53871
Perception loss distance: 0.287
Perception loss distance (ignoring rotation): 0.287
Evaluation score: 7.734

(a) Results for Delft - 10 kilometer route

(b) Results for Delft - 21 kilometer route

(c) Results for Delft - 42 kilometer route

Figure 4.5: Visualized test results for Delft


ant [53.3%]
drums [9.9%]
other
camouflage [7.8%]
Route length: 10.3 km
Relative total metric value: 2.79384
Perception loss distance: 0.534
Perception loss distance (ignoring rotation): 0.5
Evaluation score: 6.876
elephant [28.9%]
sheep [13.8%]
other
rain [13.7%]
Route length: 10.2 km
Relative total metric value: 3.89883
Perception loss distance: 0.349
Perception loss distance (ignoring rotation): 0.349
Evaluation score: 6.415
cactus [23.5%]
peas [20.3%]
other
sun [8.1%]
Route length: 10.2 km
Relative total metric value: 4.10998
Perception loss distance: 0.313
Perception loss distance (ignoring rotation): 0.313
Evaluation score: 6.92
bicycle [42.8%]
drums [19.3%]
other
ant [9.8%]
Route length: 21.1 km
Relative total metric value: 3.70316
Perception loss distance: 0.299
Perception loss distance (ignoring rotation): 0.299
Evaluation score: 5.78
elephant [56.5%]
other
frog [3%]
sheep [32.5%]
Route length: 21.3 km
Relative total metric value: 3.72586
Perception loss distance: 0.312
Perception loss distance (ignoring rotation): 0.312
Evaluation score: 5.863
hand [64.4%]
other
fire hydrant [0.9%]
cactus [32.9%]
Route length: 21.1 km
Relative total metric value: 4.31398
Perception loss distance: 0.321
Perception loss distance (ignoring rotation): 0.321
Evaluation score: 5.934
bicycle [90.1%]
other
motorbike [5.5%]
Route length: 42.6 km
Relative total metric value: 4.6135
Perception loss distance: 0.506
Perception loss distance (ignoring rotation): 0.396
Evaluation score: 6.41
elephant [75.6%]
other
ant [8.3%]
Route length: 42.3 km
Relative total metric value: 4.3497
Perception loss distance: 0.359
Perception loss distance (ignoring rotation): 0.329
Evaluation score: 5.715
hand [70.1%]
other
hedgehog [0.6%]
cactus [28.6%]
Route length: 42 km
Relative total metric value: 4.10973
Perception loss distance: 0.268
Perception loss distance (ignoring rotation): 0.268
Evaluation score: 5.109

(a) Results for Amsterdam - 10 kilometer route

(b) Results for Amsterdam - 21 kilometer route

(c) Results for Amsterdam - 42 kilometer route

Figure 4.6: Visualized test results for Amsterdam

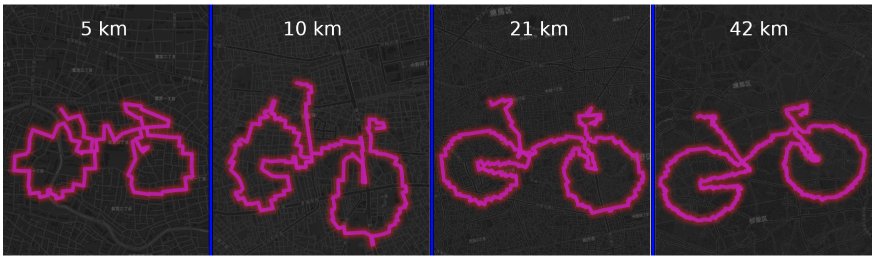


Figure 4.7: Routes of different lengths generated for the same input shape in the area of Tokyo. As the length increases, the meaning of the route becomes clearer.

straight legs, which in this case is a positive occurrence. Figure 4.8 captures the different characteristics of the same objects represented by GPS art in different cities.

The results for Delft look visually the worst. The main reason for this is the size of the city, which is comparably smaller than all the other test subjects. Its street network occupies a smaller space and the longer routes (see Figure 4.5c) lead out of the city into rural areas, where fewer roads are available. An example of an artistic route in a rural area is shown in Figure 4.9. Additionally, Delft has many small topographic barriers in the forms of canals, which cause many detours and result in more significant shape distortions of the GPS art results.

### 4.2.2 Quantitative analysis

Analyzing the numeric values from Table 4.1, the actual route length deviated from the desired length by 1.7% on average. The perception loss distance of 42-kilometer routes is generally lower than that of a 10-kilometer route, except for Delft, which can be explained by routes having to leave urban areas and take significant detours in rural roads.

When it comes to the label certainty given by the object classifier, many artistic routes have not been labelled correctly. Out of the 42-kilometer routes, only 13% of the cases were labelled inaccurately. For the 21-kilometer routes, it was 40% of the cases, and for the 10-kilometer routes, it was 80%. This is exposes the weaknesses of the object classifier. Firstly, it does not properly recognize most of the 10 kilometer routes. This is expected, because the model was not trained on GPS art data and most of these routes are significantly deformed compared to the input drawing. To overcome this issue, the neural network model should be trained at least partly using GPS art data, making it less sensitive to the jagged edges present in the routes.

Summarized results, with mean metric values per each city, are displayed in Table 4.2a. Artistic routes generated in Delft have the highest average RTMV and perceptual distance metric. These metrics have the lowest values for New York and Tokyo, which means that these cities had GPS art of the highest quality out of the tested cases. Both of them are metropolises spanning large areas with a dense road network.

Summarized results with mean metric values per each desired route length are displayed in Table 4.2b. In general, the combination of all the metrics favors longer routes, which have comparable RTMV, but generally lower perceptual distance metric. They are also more easily recognized by the object classifier, most likely because their size makes them appear

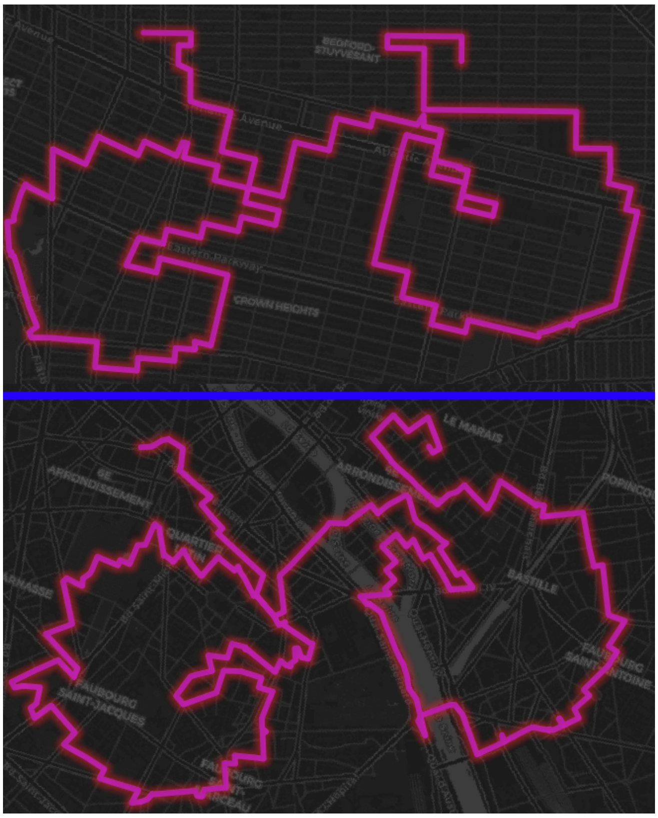


Figure 4.8: Comparison of 21-kilometer artistic routes for the same object (bicycle), generated in New York (top) and Paris (bottom).

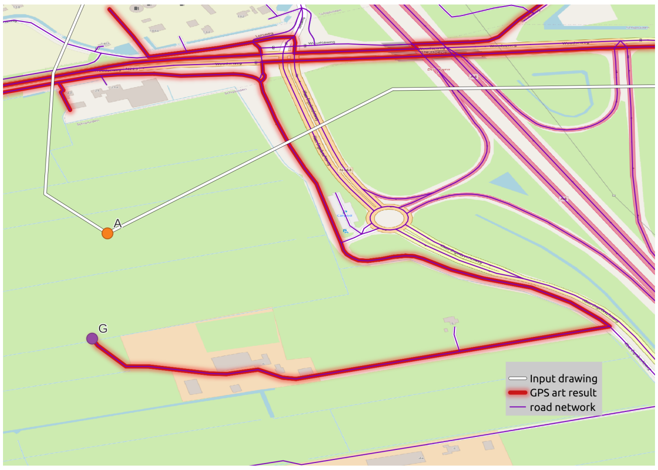


Figure 4.9: Result of the algorithm when the drawing is overlaid on top of a sparse road network. The resulting artistic route is colored red. The input drawing is a white line with black outline.

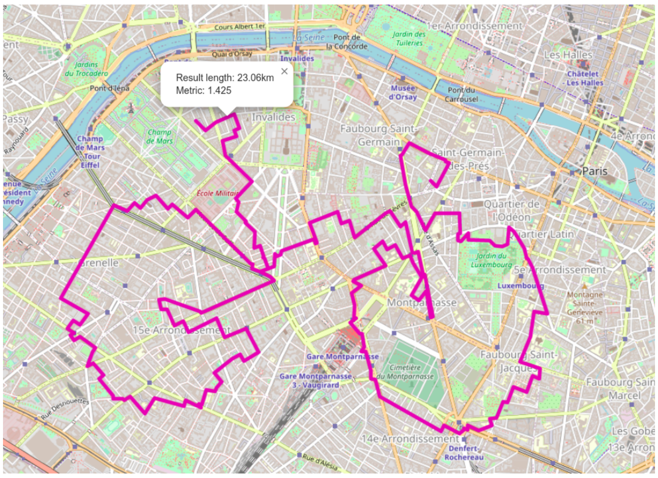


Figure 4.10: Result GPS art of a bike using the interactive approach in Paris. The river can be seen flowing north of the artistic route.

smoother than shorter routes. If there is a big enough area covered with a dense street network, then longer artistic routes will look better in most cases.

Another summary of test results in Table 4.2c, shows the average metric values when the Dutch cities (Delft, Amsterdam) are excluded. Including only the bigger cities (Paris, New York, Tokyo) causes the RTMV to drop by 46% on average, which means that the road networks in these cities allow for lower geometric deviation between the GPS art and an input shape.

## 4.3 Comparison

The test results show that the automatic GPS art workflow can be used to generate artistic routes. An alternative way would be to use the interactive web application (Section 3.5) which can achieve similar goals, but with the user's real-time interaction. At the same time, both of these methods can give unreliable results in certain cases. It is important to highlight under what circumstances these methods excel and what their pitfalls are.

One significant difference between these methods is the way in which they use the exhaustive search. The fully automatic workflow has a wider search space, with more transformation variations and more generated route candidates overall. Meanwhile, the interactive approach is limited by the desired quick response time in the user interface of the GPS art application. The exhaustive search in this case is used with a narrowed search space and is designed to function as a refinement of the route using the location picked by the user. As a result, the interactive approach explores fewer GPS art options which lowers the chance of finding the best one.

The route quality scoring system is also slightly different. To achieve faster response time, the interactive approach uses only the RTMV and the closeness of the length of the route to the desired length. The automatic workflow uses both these methods and also utilizes the more advanced, machine learning based techniques. This allows for a more comprehensive automatic evaluation of the route quality before serving the results to the user. On the other hand, the interactive approach does not necessarily need a machine learning algorithm telling the user how well it recognizes a drawing. The user is meant to do this part of the evaluation himself.

Another difference lies in the initial placement of the input drawing on the map. The automatic approach tries to determine an optimal position using template matching. In the interactive approach, the user places the drawing in an initial position according to his own preference and knowledge of the terrain. That extra knowledge of the area's topography is crucial to obtaining satisfying artistic routes in difficult scenarios, for example when the road network is sparse or there are many topographic barriers in the vicinity. The automatic workflow has no knowledge of such phenomena, therefore it cannot move its search space to where the road network is more favorable. A visual example is shown in Figure 4.10 for a route in Paris. Knowing the topographic barrier in the form of the river Seine, the user can move the drawing to a place where the route configuration is more dense. Keeping the old location would still give correct results, but in the presence of topographic barriers some of the artistic route segments may have to deviate significantly from the input drawing shape.

Comparing the automatic and interactive approaches in real world cases, the following conclusions arise. The interactive method gives better tools for dealing with difficult cases in the road layout. Additionally, it provides a way of handling user's individual preferences in real time. The automatic method finds the best of many good options in a dense road network. It explores more possibilities and uses more extensive methods of result evaluation, allowing it to find a more refined solution at the cost of lower efficiency.

## 4.4 Discussion

In this section we interpret and analyze the research findings, relating them back to the original research questions. New improvements to the existing methods are highlighted. Finally, the limitations of the solutions are discussed based on the challenges encountered in the course of this thesis.

A reminder of the previously posed research questions:
**How to automatically generate artistic routes based on any input drawing?**

Sub-questions:

- How to define measures of quality and how to evaluate them for an obtained GPS art route?
- What priorities / compromises should the designed algorithm have in order to produce optimal output considering the user's preferences?

To summarize, in this thesis we present a workflow that can automatically generate artistic routes based on certain input shapes. New methods of evaluating the quality of the generated routes were designed and implemented. The designed workflows were augmented with a number of improvements to produce optimal output, given the characteristics of the source data and specific user requirements.

### 4.4.1 Artistic route generation

The products of this thesis present two ways that artistic routes can be generated. The first one is the fully automatic GPS art workflow. It uses a combination of an image based technique (template matching) and a routing algorithm (A* algorithm) to find the solution. The other method is the semi-automatic user-centered interactive approach. It reuses some of the techniques developed for the automatic workflow, while also giving real-time feedback in the form of a responsive web application.

### 4.4.2 Measures of quality

To automatically measure the quality of artistic routes, distinct metrics were integrated into the evaluation framework. The GPS art scoring system uses a numeric description of the route's closeness to the input drawing in accordance with the metric used in the routing algorithm cost function. Considering a broader look at the aesthetic value of artistic routes, machine learning methods have been used to evaluate the quality in other aspects. An object classifier is used to check how easy it is to recognize what a route is meant to represent, while a perception loss algorithm returns a metric of visual difference between the input and the result, in an attempt to mimic human perception.

### 4.4.3 Innovation

The core part of our GPS art workflows is based on research done by Waschk and Krüger [2018]. A number of improvements have been added to their method, referred to as the core method. Additionally, the method has been integrated with supporting functionalities, like automatic quality evaluation and user-defined parameters.

#### Using A* for routing

The proposed Dijkstra algorithm for routing tasks has been replaced by A* with an euclidean distance heuristic function. This is a simple yet effective efficiency improvement, since A* is effective in start-to-end routing scenarios in street graphs. More of this is explained in Section 3.1.2.

#### Route positioning

The core method only solved the transformation-fixed approach, which assumes that the input drawing has an initial placement on the map. However, the goal was to implement a solution that can work without such a precondition for the input data. To achieve this, template matching is used to find an initial real-world location for the drawing depending on the street network layout. Then, the exhaustive search is used to further refine the drawing's position by exploring many route options and selecting the one that has the highest evaluation score.

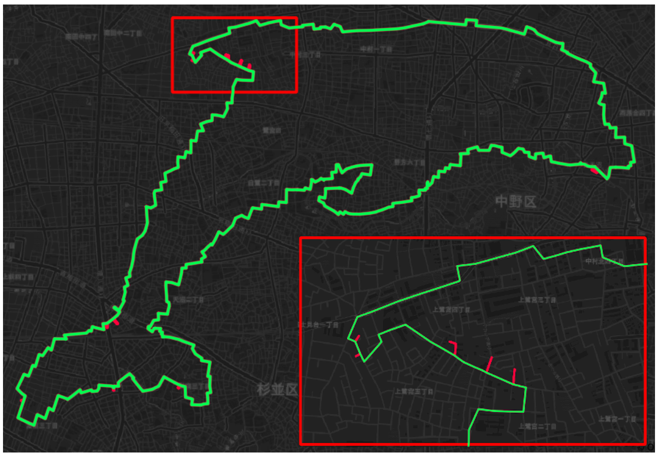


Figure 4.11: Artistic route of a dolphin (green) after removing the repeated segments (red).

#### User preferences

A significant improvement for real-world scenarios is the consideration of the user's preferences. Requirements such as a certain route length or a starting point are handled by the improved algorithm, thus making the solution more practical. The user can also express his individual preferences by using the interactive GPS art application. The instant feedback, with quickly generated and displayed artistic routes, allows for tailoring the final result according to one's needs.

#### Postprocessing

An extra postprocessing step was added to clean the artistic routes and improve their visual clarity. Repeated route segments, caused by U-turns, introduce visual irregularities. They are sometimes created by the core method as a side effect of some particular street graph configuration. The U-turns (and repeated segments) are removed, while keeping the final route's integrity as a single connected component. A visual example is shown in Figure 4.11.

### 4.4.4 Limitations

#### General

Although an image based method has been used to some extent, the final products are based on a path finding algorithm. This is because an image based approach can never provide enough information to solve a routing-based problem. The algorithm prioritizes routes that stick as close to the input shape as possible, in an attempt to recreate an input drawing in the street network. It does not consider the input shape as a whole, but rather solves a separate routing problem for each of the input shape's segments. One limitation is that the routing algorithm is not smart enough to make creative decisions. An example of such a decision could be to purposely deviate from the input drawing knowing that it will improve the GPS art quality. For example, if a person is manually designing an artistic route of an elephant, they can elongate its tusks or tail, if it is more fitting for the underlying street configuration.

When searching for the optimal artistic route, some compromises in the algorithm are introduced by the user-defined parameters. Choosing a certain starting point narrows down the search space to the routes passing via the nodes within a certain distance that point. The route length requirement also decreases the number of potential routes to choose from.

An important consideration when using any exhaustive search method is the inverse proportionality between its speed and the quality of its results. Increasing the resolution of the exhaustive search, by adding more transformation variations, allows it to explore more routing options at the cost of higher computational expense. Having more artistic route candidates increases the chance of finding the best possible route, but this comes with the mentioned trade-off.

One of the encountered limitations concerns the input data. Relating this back to the main research question, the solutions we worked on cannot generate artistic routes based on any input drawings. The designed workflow has specific requirements for the input data. Each drawing has to be represented by a single set of connected line segments. Moreover, the chosen scale of the drawing cannot be too small, otherwise it will not be possible to make a route that represents it in a meaningful way. Another data based limitation concerns the road network. As mentioned before, there are cases when the underlying street layout does not enable generating a route of sufficient quality. In an ideal case, the road network should densely cover a significant area in space.

Among other limitations is the fact that the algorithm has no knowledge of the semantic meaning of the figure. It merely follows the segments of the input shape. In this sense, it is a narrow representation of the user's idea in the road network graph. A broader approach would also consider the fact that in the end, the user is not interested in obtaining a route tightly fitted to his drawing, but a route that expresses it as clearly as possible.

### Component specific

The separate components of the GPS art workflow serve different purpose and as such, they have their own limitations. The first image matching step is based on simple template-based template matching. It compares the spatial properties of the input shape and the road network in a finite number of positions, to determine the one with the highest overlap. This method is not transformation-invariant, meaning that in its purest form it can only detect the template if its size and rotation are aligned with the pattern that is expected to be found in the target image. To overcome this, the search is repeated with the template in many variants of scale. This approach has limitations similar to those highlighted in the case of the exhaustive search method. It has a discrete search space that depends on the granularity of the used transformation variants. Because of this, its efficiency is suboptimal and the best result can be accidentally omitted. Using one of the more advanced template matching techniques may solve the mentioned issues, but it is not that straightforward.

The evaluation framework in its current form is not suitable for evaluating all kinds of GPS art. The object classifier uses a convolutional neural network, which is known to work well on inputs highly similar to the datasets used in its learning process. This means it can fail to recognize some artistic routes, if they represent something in a less conventional way that was perhaps not included in the training data. In such cases, the total evaluation score is heavily impacted by the low certainty or even false labels returned by the object classifier. Additionally, the training dataset does not share the GPS art workflow's limitation of working with only single-stroke drawings. For these reasons, the RTMV and perception loss metrics are more reliable in such cases. The former is a numeric description of the route's closeness to the input drawing. The latter is a metric of perceptual distance between any two images, in this case the input drawing and the output route. They are more robust and are not influenced by any kind of bias, as opposed to the metric that comes from the object classifier. However, they fail to capture the semantic meaning of the art, a crucial aspect which is covered by the object classifier instead.

Another consideration for the evaluation framework is related to how the total score is calculated. It is a weighted sum of its components: the RTMV, the perception loss metric and the object classifier metric. These weights should be tuned according to one's own preferences. Giving more significance to the RTMV will mean the highest rated routes will be geometrically closer to the input shape, while increasing the weight of the perception loss metric will give priority to routes that are deemed as visually closer. Tuning the weights to obtain a sufficient quality of results has to be done by trial and error.

| City | Object | Desired length [km] | Result length [km] | *RTMV* | Perceptual distance | Label certainty |
|---|---|---|---|---|---|---|
| Tokyo | Bike | 10 | 10.7 | 1.25 | 0.277 | 26% |
| | | 21 | 21 | 1.24 | 0.369 | 70% |
| | | 42 | 42.1 | 1.14 | 0.212 | 93% |
| | Elephant | 10 | 10 | 1.44 | 0.444 | N/A |
| | | 21 | 21.1 | 1.18 | 0.293 | 74% |
| | | 42 | 42 | 1.15 | 0.26 | 64% |
| | Hand | 10 | 10.3 | 1.30 | 0.301 | 50% |
| | | 21 | 21.3 | 1.08 | 0.209 | 99% |
| | | 42 | 42 | 1.29 | 0.353 | 96% |
| Paris | Bike | 10 | 10.5 | 1.78 | 0.434 | N/A |
| | | 21 | 21.1 | 1.55 | 0.399 | N/A |
| | | 42 | 41.8 | 1.93 | 0.439 | 78% |
| | Elephant | 10 | 10.6 | 1.83 | 0.47 | N/A |
| | | 21 | 21.3 | 1.53 | 0.33 | N/A |
| | | 42 | 43.7 | 1.80 | 0.299 | 66% |
| | Hand | 10 | 10.4 | 1.46 | 0.425 | N/A |
| | | 21 | 21.1 | 1.71 | 0.268 | N/A |
| | | 42 | 42.2 | 1.56 | 0.289 | 99% |

| | | | | | | |
|---|---|---|---|---|---|---|
| New York City | Bike | 10 | 9.9 | 0.98 | 0.405 | N/A |
| | | 21 | 21.4 | 1.00 | 0.34 | 59% |
| | | 42 | 41.9 | 1.60 | 0.260 | 78% |
| | Elephant | 10 | 10.2 | 1.01 | 0.477 | N/A |
| | | 21 | 21.6 | 1.11 | 0.329 | N/A |
| | | 42 | 42.4 | 1.07 | 0.230 | 66% |
| | Hand | 10 | 10.2 | 1.033 | 0.392 | N/A |
| | | 21 | 21 | 1.47 | 0.301 | N/A |
| | | 42 | 41.6 | 1.13 | 0.222 | 89% |
| Delft | Bike | 10 | 10.5 | 3.93 | 0.498 | N/A |
| | | 21 | 21.7 | 5.32 | 0.447 | 70% |
| | | 42 | 42.1 | 6.26 | 0.428 | N/A |
| | Elephant | 10 | 10.4 | 3.61 | 0.534 | N/A |
| | | 21 | 21.3 | 4.09 | 0.322 | 56% |
| | | 42 | 42.3 | 6.44 | 0.445 | N/A |
| | Hand | 10 | 9.9 | 3.03 | 0.318 | N/A |
| | | 21 | 21.2 | 3.72 | 0.301 | N/A |
| | | 42 | 42 | 6.54 | 0.287 | 61% |
| Amsterdam | Bike | 10 | 10.3 | 2.79 | 0.5 | N/A |
| | | 21 | 21.1 | 3.70 | 0.299 | 43% |
| | | 42 | 42.6 | 4.61 | 0.396 | 90% |
| | Elephant | 10 | 10.2 | 3.9 | 0.349 | 29% |
| | | 21 | 21.3 | 3.73 | 0.312 | 57% |
| | | 42 | 42.3 | 4.35 | 0.329 | 76% |
| | Hand | 10 | 10.2 | 4.11 | 0.313 | N/A |
| | | 21 | 21.1 | 4.31 | 0.321 | 64% |
| | | 42 | 42 | 4.11 | 0.268 | 70% |

Table 4.1: All test results in tabular format. For cases with an incorrect label given by the object classifier the label certainty is marked as *N/A* (not applicable).

| City | *RTMV* | Perceptual distance | Accurate labels | Label certainty |
|---|---|---|---|---|
| Tokyo | 1.23 | 0.30 | 89% | 72% |
| Paris | 1.68 | 0.37 | 33% | 81% |
| New York | 1.16 | 0.33 | 44% | 73% |
| Delft | 4.77 | 0.40 | 33% | 62% |
| Amsterdam | 3.96 | 0.34 | 78% | 61% |

(a) Summary of results. Metric values are average per city.

| Desired route length | *RTMV* | Perceptual distance | Accurate labels | Label certainty |
|---|---|---|---|---|
| 10 | 2.23 | 0.41 | 20% | 35% |
| 21 | 2.45 | 0.32 | 60% | 66% |
| 42 | 3.00 | 0.31 | 87% | 79% |

(b) Summary of results. Metric values are average per desired route length.

| Desired route length | *RTMV* | Perceptual distance | Accurate labels | Label certainty |
|---|---|---|---|---|
| 10 | 1.34 | 0.4 | 13% | 38% |
| 21 | 1.32 | 0.32 | 27% | 76% |
| 42 | 1.41 | 0.28 | 60% | 81% |

(c) Summary of results without the Dutch cities (Delft, Amsterdam). Metric values are average per desired route length.

Table 4.2: Summary of test results

# 5 Conclusions

This chapter summarizes main findings and conclusions of this thesis. Potential tasks for future work are also described.

## 5.1 Summary

In this thesis we present work in the topic of designing and implementing a solution for generating artistic routes from input drawings. The results include two routing-based products that can be used for the mentioned purpose: the automatic GPS art workflow and the interactive GPS art application. The presented solutions are capable of generating shape-guided artistic routes when certain conditions regarding the input and the road network data are met. Each of the solutions has its own advantages and limitations. The automatic GPS art workflow, supported by a comprehensive evaluation framework, excels in finding the best routes out of many possibilities in a dense road network. It requires more time to generate all the options, evaluate them and choose the one with the highest score. The interactive application is more user-focused and gives better tools for dealing with more difficult scenarios, for example when the road network has more topographic barriers (farm fields, rivers, railways). Additionally, it has some efficiency adjustments and is more pleasing to use with its instant feedback in the form of displayed routes. As a trade-off, it only uses a limited capability of the designed evaluation framework.

Test results prove that GPS art is generally not well-suited for short routes. As the route length increases, its representation of the source drawing also becomes more visually pleasing. Another limitation is encountered in rural areas, where there are fewer routing options, because of lower street density.

As a final consideration, it should be mentioned that the quality of GPS art is difficult to measure. For the complex real world cases, a reference solution that is 100% identical to the input drawing almost never exists. Because of this, there is a permanent uncertainty, whether the best obtained result is actually the best within the searched road network.

## 5.2 Future work

In the course of this thesis we discovered many new paths towards the improvement of the solution. The GPS art workflow in its current form can be made smarter and more efficient. The routing component is irreplaceable, since it is crucial to obtaining the final solution as a subset of the road network graph. Other components and auxiliary tools can be developed to further improve the GPS art workflow.

Future work could be more focused on preserving the semantic meaning of the art, instead of only trying to preserve its geometric properties. There can be many potential ways to move closer towards this goal. One of them would be to recognize what the user wants to express in the first place. Then, an algorithm could be used to generate semantically meaningful deformations that would lead to exploring new artistic route options. Semantically meaningful deformations are understood as changes which modify a drawing's geometry without changing its meaning. An example of such a deformation would be elongating an elephant's tusks or its tail. Designing such an algorithm would be an innovative challenge. Similar research, although without the semantic meaning aspect, has been done by Igarashi et al. [2005].

Another idea is to design an application that stores many variations of semantically labelled drawings. After learning the kind of the object that the user wants to express, the application could suggest many potential routes that can express the user's idea in distinct ways.

Another point for future work is related to the template matching step in the GPS art workflow. In our methods, we only used the simpler template-based template matching technique, which is not invariant to scale and rotation. Successfully applying a feature-based technique will result in faster and more robust pattern detection in the road network, but it is not without challenges. The features used in most existing methods are designed for texture-rich images with color information, and are therefore not effective for the textureless, binary images that are derived from geometric shapes. Kim and Araújo [2007] describe a method which can be further explored to make it work for the GPS art case.

An alternative method of template matching proposed by Lee et al. [2016] enables finding similar objects, even when the deformation between the source and target is not due to an affine transformation. Finding a way to elastically match input shapes to the patterns of the road network could bring a revolution in the GPS art problem.

The template matching step could also be replaced with a different technique for pattern detection in images. More research needs to be done to check if learning-based object detection can give better results in locating shapes in the road network. Challenges similar to those described in the context of feature-based template matching are expected to lie ahead. The pattern detection problem could also be treated as a shape matching problem. Takase and de Sales Guerra Tsuzuki [2009] present a method for matching 2-dimensional shapes which is invariant under translation, rotation and scaling transformations of the compared shapes.

The learning-based object classifier has room for improvement, which was shown in the tests (Section 4.2). The model that we used was a generic one, meant for doodle sketch recognition. With some adjustments in the learning datasets, a new model for the GPS art case can be designed to provide better object recognition for the evaluation framework.

## 5.3 Reflection

Throughout the course of this thesis, we have encountered many challenges which enabled us to gain new knowledge in a variety of topics. The goal of making automatic GPS art generation possible required focusing on software development. As highlighted in Table 3.2 and Table 3.3, we ended up using a broad technological stack consisting of multiple programming languages (C++, Python, Javascript) and software packages for the different tasks. Some of the used software libraries required a significant input of time to be able to use them efficiently and tailor their functionality to the challenges of this thesis. A significant amount of time was spent on figuring out various C++ libraries like Boost (graph data types, routing) and CGAL (geometry processing, nearest neighbor search). We were also able to improve the knowledge of graph theory, routing algorithms and image matching techniques.

Another challenge was related to the quality of the road network data. The used datasets, while freely available, had their flaws. Disconnected components, often present in the road network, complicated the work, because extra time had to be spent cleaning up the graph. Maintaining a valid, up-to-date and production-ready routing data is a service done commercially by specialized companies.

We knew from the beginning that automatically generating GPS art may not have a flawless solution, due to the nature of the problem. We did our best to explore different paths that could lead to satisfactory results. Still, the visual quality of GPS art is often insufficient in cases where the data or the specific user requirements do not allow for better options.

After working with GPS art for a long time, we developed a bias allowing us to recognize GPS art more easily, but causing our perception of the art's quality to become more subjective. We leave the judgment of visual satisfaction with the presented results up to the reader of this thesis. After all, beauty is in the eye of the beholder.

# Bibliography

Balduz, P. (2017). Walk line drawing.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Brunelli, R. (2008). Template matching techniques in computer vision: Theory and practice.

Igarashi, T., Moscovich, T., and Hughes, J. F. (2005). As-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, 24(3):1134–1141.

Kim, H. and Araújo, S. (2007). Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast. pages 100–113.

Kim, J. and Shontz, S. M. (2010). An improved shape matching algorithm for deformable objects using a global image feature. In Bebis, G., Boyle, R., Parvin, B., Koracin, D., Chung, R., Hammound, R., Hussain, M., Kar-Han, T., Crawfis, R., Thalmann, D., Kao, D., and Avila, L., editors, *Advances in Visual Computing*, pages 119–128, Berlin, Heidelberg. Springer Berlin Heidelberg.

Lee, H., Kwon, H., Robinson, R. M., and Nothwang, W. D. (2016). Dtm: Deformable template matching.

Lei, R., Yang, B., Quan, D., Li, Y., Duan, B., Wang, S., Jia, H., Hou, B., and Jiao, L. (2021). Deep global feature-based template matching for fast multi-modal image registration. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 5457–5460.

Networks, U. R. (2016). Urban road network data. `https://figshare.com/articles/dataset/Urban_Road_Network_Data/2061897`.

Rachmawati, D. and Gustin, L. (2020). Analysis of dijkstra's algorithm and a* algorithm in shortest path problem. *Journal of Physics: Conference Series*, 1566(1):012061.

Santoso, L. W., Setiawan, A., and Prajogo, A. K. (2010). Performance analysis of dijkstra, a* and ant algorithm for finding optimal path case study : Surabaya city map.

Takase, F. K. and de Sales Guerra Tsuzuki, M. (2009). Translation, rotation and scaling invariant 2d shape matching based on filter matching. *IFAC Proceedings Volumes*, 42(4):1724–1729. 13th IFAC Symposium on Information Control Problems in Manufacturing.

Waschk, A. and Krüger, J. (2018). Automatic route planning for gps art generation. In *SIGGRAPH Asia 2018 Technical Briefs*, SA '18, New York, NY, USA. Association for Computing Machinery.

Wiki, O. (2022). About — openstreetmap wiki,. `https://wiki.openstreetmap.org/w/index.php?title=About&oldid=2368272`. [Online; accessed 15-January-2023].

Zhang, L. (2021). Hand-drawn sketch recognition with a double-channel convolutional neural network. *EURASIP Journal on Advances in Signal Processing*, 2021.

Zhang, R., Isola, P., Efros, A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric.

## Colophon

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.