TU WIEN Informatics

Embedding User-Defined Shapes into Metro Map Layouts

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Medieninformatik und Visual Computing

eingereicht von

Tobias Batik
Matrikelnummer 11701221

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ph.D. Hsiang-Yun Wu

Wien, 21. März 2021

Tobias Batik | Hsiang-Yun Wu

Embedding User-Defined Shapes into Metro Map Layouts

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Media Informatics and Visual Computing

by

Tobias Batik
Registration Number 11701221

to the Faculty of Informatics

at the TU Wien

Advisor: Ph.D. Hsiang-Yun Wu

Vienna, 21st March, 2021

Tobias Batik   Hsiang-Yun Wu

Erklärung zur Verfassung der Arbeit

Tobias Batik

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 21. März 2021

Tobias Batik

Danksagung

Ich möchte vor allem meiner Betreuerin Ph.D. Hsiang-Yun Wu für ihre Zeit und umfangreiche Unterstützung während dieses Projektes, sowohl als auch für die Bereitstellung der Grundstruktur des Codes und Daten danken.

Weiters möchte ich Lucie Jo Knilli für ihre Unterstützung, sowie für das Korrekturlesen danken.

Acknowledgements

I would especially like to thank my supervisor Ph.D. Hsiang-Yun Wu for taking extensive time to support me throughout this project; providing me the basic code structure and metro network data; as well as giving further input.

Additionally I would also like to thank Lucie Jo Knilli for her encouragement as well as taking her time to proofread.

Kurzfassung

U-Bahn Pläne sind schematische Repräsentationen, mit dem Ziel Orientierung und Navigation in U-Bahn Systemen zu vereinfachen. Diese Pläne zu erstellen ist allerdings relativ komplex. Aus diesem Grund wurden in den vergangenen Jahren einige Algorithmen entwickelt, um diese Darstellungen automatisch generieren zu können. Jedoch hat bei diesen Herangehensweisen der*die Designer*in nur eingeschränkte Möglichkeiten das generierte Layout zu beeinflussen und kontextuelle Informationen der Stadt können nicht berücksichtigt werden. Um diese Limitierungen umgehen zu können, wird in dieser These eine Methode vorgestellt, bei der potentielle User*innen die Möglichkeit haben das resultierende Layout mit Hilfe von Leitpfaden zu beeinflussen. Dieser Ansatz kann genutzt werden, um optisch ansprechende U-Bahn Pläne zu generieren, sowie symbolische Formen mit Hilfe von U-Bahnlinien nachzubilden. Das "mixed-layout"– manche Kanten werden rotiert um parallel zum naheliegendstem Leitpfad zu sein und andere Teile des Netzes werden dazu gebracht octolinear zu verlaufen – wird eingeführt, um die Leitpfade besser in das Layout integrieren zu können. Um die Potentiale dieser Methodik skizzieren zu können, werden beispielhaft einige U-Bahn Netze generiert und anschließend diskutiert.

Abstract

Metro maps are essential when navigating through a public transportation network. They are schematic maps that have the aim to make orientation and navigation through a metro system easier. But creating them is quite complicated. Therefore numerous algorithms have been developed in the past trying to generate these maps automatically. The downside to using this approach is, that the designer only has limited possibilities to influence the resulting layout as well as contextual information of the city can not be taken into account. In order to overcome this limitation, this thesis presents a method where a potential user could influence the resulting layout by adding a set of guide paths. This method can be used to create artistically pleasing metro maps as well as make metro lines follow symbolic shapes in the layout. A mixed-layout – where some edges are rotated to be parallel to the closest guide path and other parts are octilinear – is proposed to integrate the guide paths better into the layout. To outline the potentials of this approach, examples of several metro networks were generated and are later also discussed.

Contents

CHAPTER 1

Introduction

The first chapter captures the motivation behind this project and points out the importance of metro maps. The goal for this project as well as how this thesis is contributing to the overall research in this field is outlined in sections 1.3 and 1.4. In section 1.5 the structure of this thesis is explained.

1.1 Background

When navigating through urban centers metro maps play an essential part. They are simplified as well as optimized schematic representations of often quite complex metro networks, allowing one to quickly find stations, intersections and plan routes across the entire city network. Therefore numerous layouts have been developed within the last decades – some of them being more restrictive than others. The most widely used style is called octilinear, which was first used in 1931 for the Berlin S-Bahn and shortly after also for the famous London metro maps developed by Henry Back [Ove15].

In layouts like these, all edges connecting two stations are revolved to an angle of a multiple of $\frac{\pi}{4}$. In some of today's used metro map layouts the restrictions regarding the allowed angle of edges are loosened for parts of the map – an example being metro lines that run in the shape of an irregular circle that are then displayed perfectly circular. A good example for a mixed layout like that is the Moscow metro map [Met19], which consists of one – in some versions two – lines shaping a circle around the city center.

1.2 Motivation

Metro maps with a restrictive layout – like octilinear – are often complicated to produce. Designing a schematic representation by hand as such is a challenging and very time-consuming task. Therefore numerous layout algorithms and computational approaches have been developed in the past [WNT+20]. These approaches automatically create a schematic representation of the metro system based on the geographic positions of the metro stations. State-of-the-art approaches create commonly good results. But naturally, these maps have numerous downsides compared to ones made by hand. Some of the downsides being, that potential designers have very little control over the resulting layout; contextual information about the city, which is not encoded in the metro network can not be considered when creating the layout; and to the best of the authors' knowledge, up until now no approach has been developed that enables metro map layouts where some edges can be rotated to arbitrary and others to octilinear angles, based on suggestions of a designer.

1.3 Goal

Further, the guide paths should also be used for creating mixed layouts. Therefore a subset of edges in the metro network should rotate in order to run parallel to the closest guide path and the other edges should rotate to an octilinear angle. The suggested process should help generate visually interesting results and fulfill the needs of a modern metro map.

1.4 Contributions

The contributions of this thesis are summarized as:

- A method is presented for creating metro map layouts that are influenced by guide paths.
- With the suggested process, metro layouts can be calculated, where a subset of edges are rotated to run parallel to the guide paths and the rest of the network is arranged in an octilinear layout.
- An interactive framework is developed, where the user can iteratively create metro networks with different settings and variable configurations.

1.5 Structure

Chapter 2 is a summary of previous work that is relevant for the process of this thesis. Chapter 3 outlines the methodology used to generate a mixed metro map layout. The implementation of the suggested method is summarized in chapter 4 and the results are then presented in chapter 5, where then also the quality of the results, as well as the limitations are discussed. In chapter 6 a summary of the process and work is given and potential future work is discussed.

CHAPTER 2

Related Work

This chapter outlines and summarizes previous works that are relevant for this thesis.

2.1 Layout Types

In the past decades, multiple layout strategies have been developed. One of them being the octilinear layout where the angles of the edges are restricted the to a multiple of $\frac{\pi}{4}$. There are other layout variations that allow more or less angle possibilities – for example having instead of four angles six. Maps drawn in a multi-linear style follow a less restricted layout, where the edges can be oriented in all possible directions [RGL17]. In curve-linear approaches the metro lines are shaped as smooth curves rather than straight lines. This approach aims to increase the flow of the map and tries to avoid large changes in metro line curvature [RGL17].

Figure 2.1 right shows the current Moscow metro map, which is an example for a octilinear layout – here only two metro lines are not arranged octilinear. The historic Moscow metro map, shown in Figure 2.1 left, is an early example of a multi-linear layout.

Roberts [Rob14] conducted an internet survey where nine different variations of the London metro map are rated by subjects. The tested maps varied based on design rules (e.g. layout type) and design priorities. Roberts outline that straight lines improve the usability of maps and that octilinear layouts perform the best when it comes to usability. Also other publications emphasize the advantages of this layout [RGL17] [RV16]. But Roberts et al. also argue that there is no single set of rules that can be used for all metro maps and therefore one needs to make design criteria according to the network that is used and the desires one has [RNL+13].

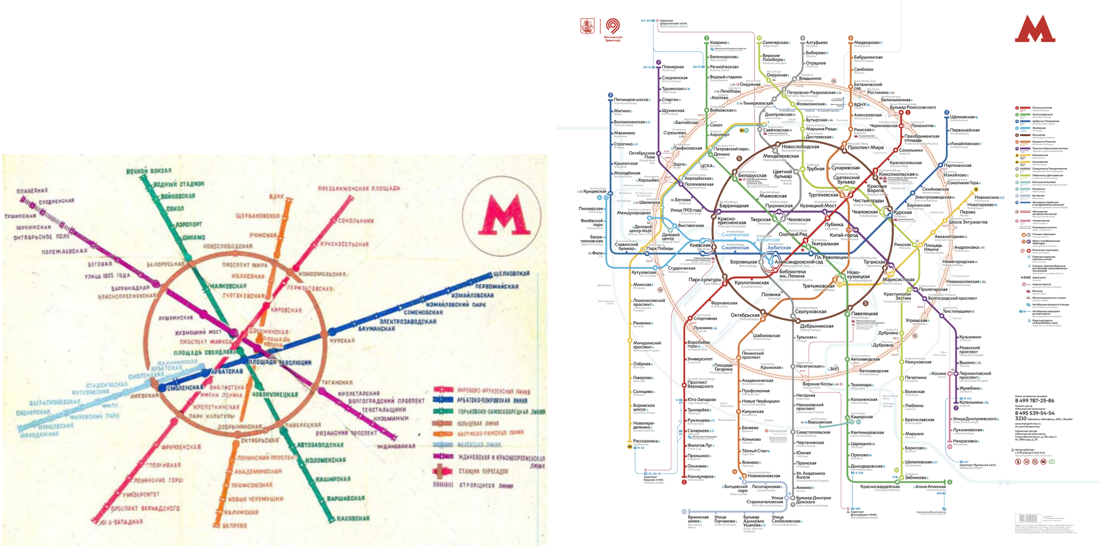

Figure 2.1: Official metro maps of Moscow. Left: the official map of 1974 [Ove15]; Right: the metro map that is currently in use [Met19].

2.2 Previous Approaches

Generally, there are two difficulties that appear when generating a metro map. One being the layout problem, where the metro lines are transformed and simplified, and the other one being how to place labels [HMdN06, NW10]. This thesis only targets the difficulty mentioned first.

Based on a set of official maps of public transportation, Nöllenburg and Wolf [NW10] have defined seven design criteria and five of them target the metro layout problem. Numerous layout algorithms have been developed with the aim to fulfill these rules – Wang et al. [WC11] and Onda et al. [OMI18] being some of them. Further Wu et al. [WNT+20] has outlined the use of symbolic shapes for metro lines of specific typology to potentially increase the legibility of the metro map.

Numerous layout algorithms for generating metro maps have been developed within the last few years – the approach by Nöllenburg and Wolff [NW10] being the one that is often referred to as the one giving very good results. They used mixed integer programming to calculate an octilinear layout as well as the labels of the stations simultaneously. The downside of the process they used is that it takes a lot of time to generate a map and therefore this algorithm is not handy when working interactively. This approach was further developed and generalized by Nickel et al. [NN20].

Hong et al. [HMdN06] and Chivers et al. [CR14] presented methods for generating octilinear layouts with a force-based approach. Further, Fink et al. [FHN+12] presented an approach for generating curve-linear layouts, where metro lines are shaped by Bezièr curves. Numerous other approaches have been developed in the past – all having their own advantages and disadvantages [SR05, SRMOW10, NRW19, OMI18].

The work of Wang et al. [WC11] introduces a more efficient and less time-consuming process for calculating octilinear layouts, compared to for example Nöllenburg and Wolf's approach. In their publication they outline a two-step process for transforming the geographically accurate representation of the metro network into a simplified version. First, a smooth, curve-linear layout is generated, for which they defined a number of local constraints that apply to all vertices and edges of the graph. These constraints are minimized in an iterative process. In a second step, the curve-linear layout is transformed to be octilinear. For this the edges are rotated to an angle of a multiple of $\frac{\pi}{4}$. The approach by Wang et al. produces high-quality metro maps within a relatively short time. The approach presented in this thesis builds upon the algorithm and constraints defined by Wang et al. and extends their work. Further, Wang and Peng [WP15] developed an approach where the user can influence the resulting layout by manipulating a few control handles. Their system performs in real-time enabling interactive map editing and therefore human knowledge can also be taken into account when designing a metro map.

Lin et al. [LLHL14] outlines an algorithm that deforms dense road networks in order to include user-defined paths. An underlying grid structure of the road network is deformed based on several geometric and aesthetic constraints. Pan et al. present a method in their work [PCZ$^+$20] where the user first defines a layout for a sub-graph and after doing so, the layout of the topologically similar subgraphs are modified based on the user's layout. Similar to the process introduced in this thesis, the target layout structure is translated, rotated and scaled based on the matching that is done before the subgraph is deformed.

There are several approaches that quantify the similarity between two shapes. One example being the work of Latecki et al. [LL00]. But quantifying the recognizability of a shape inside a layout is rather difficult. If a user is able to identify the integrated guide shape is based on multiple factors. Among others being how iconic or complex the shape is. Further, the cultural background of the reader could also influence the recognizability of certain shapes. To the best of the author's knowledge, until now, no approach has been developed than enables calculating the similarity between a guide shape and a loosely defined sub-graph.

CHAPTER 3

Methodology

In this chapter firstly a definition of the design objectives of modern metro maps is given, as well as additional design criteria are outlined targeting the aims of this project. A brief overview of the process of creating mixed metro layouts is given and in section 3.3 the used notation is described. A number of local as well as global constraints are outlined in sections 3.4, 3.5 and 3.6. How these constraints are applied and minimized to fulfill the described design criteria is elaborated in section 3.7.

3.1 Design Criteria

This section outlines design rules that aim to formulate objectives of a modern metro map layout. Nöllenbur et al. [Nöl14] and other publications [WNT+20, WC11] outlined a number of design criteria which aim to increase the readability and usability of metro networks – the most important criteria for the process of this thesis are summarized below.

Uniform Edge Lengths In metro maps, the lengths of the edges between two stations are as uniformly as possible [NW10]. This eliminates the information about distances from the map, but in general it creates a more clear and grid like layout. Wang et al. [WC11] argued that travel time in urban metro networks depends more on the number of stations rather than the actual traveled distance.

Straight Metro Lines Metro lines should run in as straight lines as possible [NW10, Rob14], because that makes it easier to track them along the map.

Large Angles The angles between two connected edges should be as big as possible [NW10]. Therefore, in the case of an intersection between different metro lines, the edges are evenly spaced around the station, so that it is easier to distinguish them.

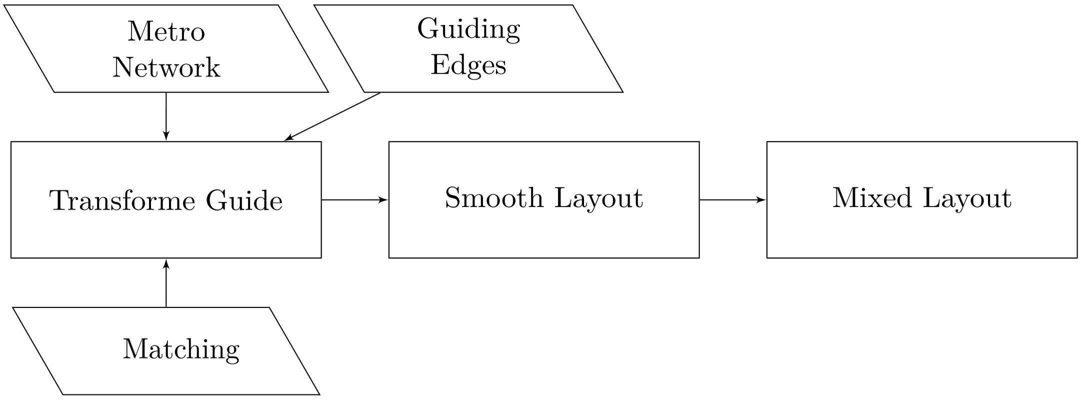

Figure 3.1: Overview of the process of creating the mixed layout.

Relative Station Position The relative position between two stations should be preserved. Otherwise this could create a conflict with the mental map of the city that residents have [NW10].

Limited Edge Orientations As usual in modern metro maps, restrict edges to a small number of allowed angles [NW10]. Most common to a multiple of $\frac{\pi}{4}$ [Rob14].

Minimum Distance Between Stations and Edges The distance between a metro station and a not connected edge should not be to small [WNT+20]. Otherwise this could indicate an interchange which does not exist in the real-world metro network. Further the length of an edge should not be less than a minimum [NW10].

The aforementioned design rules have been defined in the past and apply to metro maps in general. Additionally two further design criteria are formulated for this project. These criteria describe the desired affect of the guide paths on the final layout.

Guide Representation The metro layout should be influenced by the guide paths so that the metro lines partially align along the shapes of these guide paths.

Mixed Edge Orientation Edges of the metro network that are replicating the guide path should be oriented in the same direction as the closest section of the guide path. Other edges should follow the edge orientation design principle outlined above. So the edges possible orientation is restricted to a multiple of $\frac{\pi}{4}$.

3.2 Generating Mixed Metro Maps

The mixed metro map layout is created by deforming the geographically accurate representation of the metro network. In order to influence the outcome of the resulting layout a set of guiding edges is taken into account. These guiding edges enforces metro

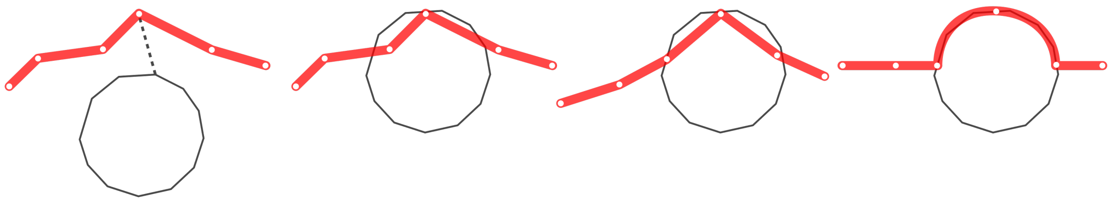

Figure 3.2: Calculating mixed layout; From left to right: initial layout with matching, translated guide, smooth and mixed layout.

stations to move towards them and edges between metro stations are rotated so that they are parallel to the closest guide edge. Furthermore, the guide edges are also used to determine, if an edge between two stations should be rotated to be octilinear or not.

The algorithm starts with a geographically accurate representation of the metro network and a set of guiding edges. In order to indicate the position of the guiding edges, a number of matches between metro stations and vertices of the guiding edges must be defined.

The final mixed layout is calculated in three steps. An overview of the pipeline can be seen in Figure 3.1.

Step 1 The guiding edges are globally translated, scaled, and rotated to fit the initial metro layout. This transformation is based on a matching between metro stations and vertices of the guide path.

Step 2 A smooth layout is calculated. This is influenced by the guide paths.

Step 3 The final mixed layout is created by rotating edges either to be octilinear or parallel to a guiding edge.

The three steps are illustrated in Figure 3.2. In order to create the smooth and the mixed-layout, local constraints are defined. Additional global constraints are formulated to rotate, translate and scale the guiding edges. The layout is created by an optimization process where these constraints are minimized.

3.3 Notation

Because metros commonly operate in both directions, there is no need for directed edges and therefore an underacted graph can be used when creating metro maps. Each vertex of the graph represents a station and each edge represents a connection between these stations.

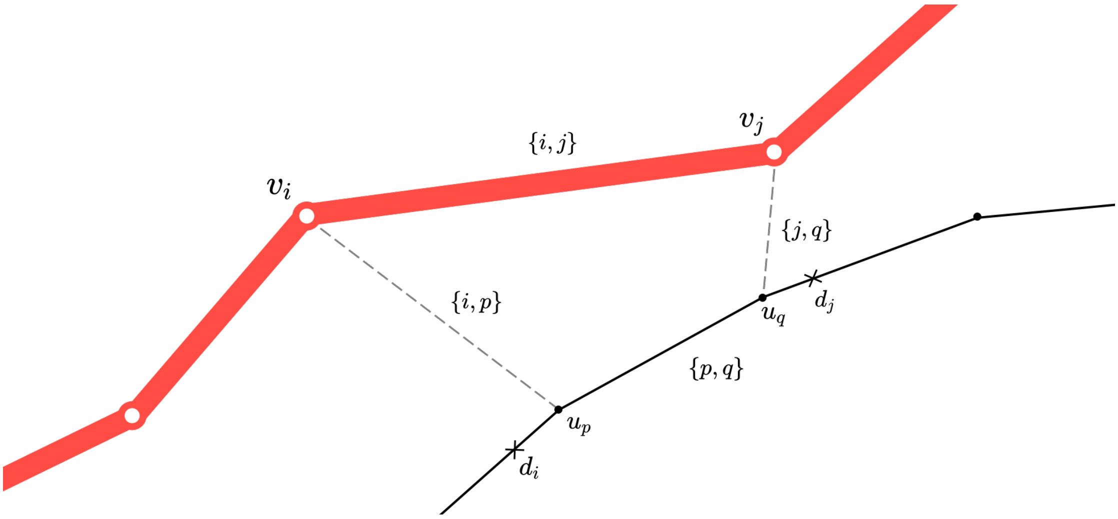

Figure 3.3: The red line indicates a metro line. Metro stations are represented as white dots and revert to as $v_i, v_j \in V_{\text{metro}}$. The edge $\{i,j\} \in E_{\text{metro}}$ represents the connection between two stations v_i and v_j. The black line represents a set of guiding edges. Vertices of the guiding graph $G_{\text{guide}} = (V_{\text{guide}}, E_{\text{guide}})$ are denoted by $u_p, u_q \in V_{\text{guide}}$. The matching between guide vertices and metro vertices are displayed as dashed gray lines. The closest point of a metro vertex $v_i \in V_{\text{metro}}$ on a guide edge is marked with a black cross and described with d_i.

The initial layout – which is based on geographical coordinates of the stations – is denoted by the graph $G_{\text{metro}} = (V_{\text{metro}}, E_{\text{metro}})$. $V_{\text{metro}} = \{v_1, v_2, ..., v_n\}$ is the set of metro vertices. Each element $v_i \in V_{\text{metro}}$ represents a single station and $v_i = (v_{i,x}, v_{i,y}) \in \Re^2$ is a vertex with the initial positions of the metro graph. Edges between two vertices are denoted by $\{i,j\} \in E_{\text{metro}}$. Edges $\{i,j\} \in E_{\text{metro}}$ are also referred to as the edges of the metro lines. $G'_{\text{metro}} = (V'_{\text{metro}}, E'_{\text{metro}})$ is the graph of the smooth layout and the mixed layout is written as $\tilde{G}_{\text{metro}} = (\tilde{V}_{\text{metro}}, \tilde{E}_{\text{metro}})$. The graph that represents the guiding edges is denoted by $G_{\text{guide}} = (V_{\text{guide}}, E_{\text{guide}})$ where $u_p \in V_{\text{guide}}$ is a vertex of this graph. $G_{\text{guide}} = (V_{\text{guide}}, E_{\text{guide}})$ is also a guiding path. Edges are defined as $\{p,q\} \in E_{\text{guide}}$ and referred to as guiding edges. The guiding graph that is transformed to fit the initial metro map is denoted by $G'_{\text{guide}} = (V'_{\text{guide}}, E'_{\text{guide}})$.

The set of matches between metro stations and guiding edges is denoted by E_{match}. Where the matching between a station and a vertex of the guide graph is represented by an edge between these vertices. These edges are defined as $\{i,p\} \in E_{\text{match}}$. Where $v_i \in V_{\text{metro}}$ is a metro station and $u_p \in V_{\text{guide}}$ a guide vertex. The spatially closest point on a guiding edge $\{p,q\} \in E_{\text{guide}}$ to a vertex $v_i \in V_{\text{metro}}$ is denoted by d_i. An illustration of the notation can be seen in Figure 3.3. The transformation of the geographically accurate metro map into a mixed layout metro map based on the guiding edges and the matching can be described with the following function: $(G_{\text{metro}}, G_{\text{guide}}, E_{\text{match}}) \rightarrow (\tilde{G}_{\text{metro}}, G'_{\text{guide}})$.

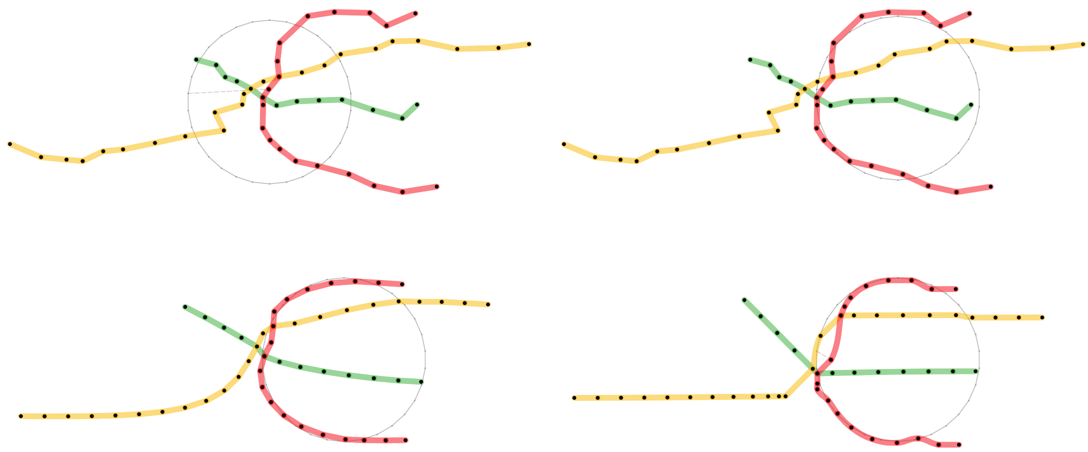

Figure 3.4: Process of creating the mixed layout. The Figure at the top left shows the initial layout of the metro, the guide path, as well as a matching between two vertices (displayed with the dotted line). The Figure located at the top right shows the optimized guide position. The Figure at the bottom left displays the smooth layout, and the Figure at the bottom right the final mixed layout including curved edges.

3.4 Guide Transformation Constraints

The resulting metro map depends on the spatial position of the guiding edges G'_{guide}. Therefore in the first stage a potential user defines the set of edges $\{i, p\} \in E_{\text{match}}$ based on which the graph G_{guide} is globally translated, rotated and scaled in order to minimize the equation:

$$\Omega_{\text{transform}} = w_{\text{translate}}\Omega_{\text{translate}} + w_{\text{rotate}}\Omega_{\text{rotate}} + w_{\text{scale}}\Omega_{\text{scale}}. \tag{3.1}$$

In order to preserve the overall shape, the guide is only globally transformed. Weights are denoted by w. The translation constraint $\Omega_{\text{translate}}$ is given by the sum of the magnitude of all edges $\{i, p\} \in E_{match}$. This can be formalized as:

$$\Omega_{\text{translate}} = \sum_{\{i,p\} \in E_{\text{match}}} |u_p - v_i|^2. \tag{3.2}$$

Where $u_p \in V_{\text{guide}}$ is a vertex of the guide graph and $v_i \in V_{\text{metro}}$ of the metro graph. If $\|E_{\text{match}}\| = 1$ an ideal solution can be found by just minimizing $\Omega_{\text{translate}}$. If $\|E_{\text{match}}\| = 2$ an ideal solution can be found by translating, rotating and scaling the guide graph. But if $\|E_{\text{match}}\| > 2$, no ideal solution can be guaranteed. By rotating the guide G_{guide}, the constraint Ω_{rotate} and by scaling, the constraint Ω_{scale} can be minimized.

The rotation constraint is given by:

$$\Omega_{\text{rotate}} = \sum_{\{i,p\},\{j,q\} \in E_{\text{match}}} \min(|\theta_{ij} - \theta_{pq}|, |\theta_{ij} - \theta_{pq} + \pi|)^2 \qquad \text{for } \|E_{\text{match}}\| > 1, \tag{3.3}$$

where θ_{ij} is the angle of $|v_i - v_j|$ and θ_{pq} the angle of $|u_p - u_q|$. The scaling constraint Ω_{scale} can be defined as:

$$\Omega_{\text{scale}} = \sum_{\{i,p\},\{j,q\} \in E_{\text{match}}} |\|v_i - v_j\| - \|u_p - u_q\||^2 \qquad \text{for } \|E_{\text{match}}\| > 1, \tag{3.4}$$

where $u_p, u_q \in V_{\text{guide}}$ are guide vertices and $v_i, v_j \in V_{\text{metro}}$ are metro vertices.

3.5 Smooth Layout Constraints

The smoothing stage aims to fulfill two goals: First, the resulting metro map should have the qualities of a modern metro map, as described in section 3.1 and secondly, the metro network should integrate the guiding edges $\{p, q\} \in E'_{\text{guide}}$ as well as possible. For example, if the guiding edges E'_{guide} are in the shape of a heart, the heart should (still) be recognized by a viewer even when the guiding edges are not displayed. In order to achieve this effect, two key observations are of importance: The shape is well represented if metro stations are placed with minimum distance to a guiding edge and metro edges should be parallel to the spatially closest guide edge.

To create the smooth layout the constraint Ω_{smooth} is minimized. This constraint can be summarized as:

$$\begin{aligned} \Omega_{\text{smooth}} = \quad & w_{\text{close}}\Omega_{\text{close}} + w_{\text{par}}\Omega_{\text{par}} + w_{\text{over}}\Omega_{\text{over}} + \\ & w_{\text{len}}\Omega_{\text{len}} + w_{\text{max}}\Omega_{\text{max}} + w_{\text{pos}}\Omega_{\text{pos}}. \end{aligned} \tag{3.5}$$

Where as Ω_{len}, Ω_{max} and Ω_{pos} are used to create a map layout with uniform distances between adjacent vertices and metro lines that are as straight as possible. These constraints have initially been defined by Wang et al. [WC11]. In addition the three constraints Ω_{close}, Ω_{par} and Ω_{over} have been defined. By minimizing these, the metro map is transformed based on the guiding edges. Weights are denoted by w. Constraints of higher importance should be weighted heavier than the ones that are not as crucial.

3.5.1 Close Constraint

The constraint Ω_{close} aims to translate vertices $v_i \in E'_{\text{metro}}$ in the direction of the closest guiding edge $\{p, q\} \in E'_{\text{guide}}$. Ω_{close} is defined as:

$$\Omega_{\text{close}} = \sum_{v'_i \in V'_{\text{metro}}} |(v'_i - d'_i)t_{id}|^2 \tag{3.6}$$

$$\text{were} \quad t_{id} = (\frac{\deg(v_i)}{2})^3 \min(1, \frac{1}{\|v'_i - d'_i\| C_s}).$$

$\deg(v_i)$ is the number of outgoing edges of the vertex $v_i \in V'_{\text{metro}}$ and C_s is a constant. t_{id} is formulated so that vertices with a higher degree are weighted more heavily. Early experiments showed that this creates more clean layouts. d_i is the spatially closest point on the closest edge $\{p,q\} \in E'_{\text{guide}}$ to v'_i. This point is denoted by

$$d'_i = u'_p + ((v'_i - u'_p) \cdot (u'_q - u'_p))(u'_q - u'_p), \tag{3.7}$$

where $\cdot$ denotes the dot product between two vectors.

3.5.2 Parallel Constraint

As previously described, in order to represent the guiding shape as well as possible in the smooth metro layout G'_{metro}, metro edges $\{i,j\} \in E'_{\text{metro}}$ should be rotated towards the closest guide edge $\{p,q\} \in E'_{\text{guide}}$. Therefore the angle between $\{i,j\} \in E'_{\text{metro}}$ and $\{p,q\} \in E'_{\text{guide}}$ should be minimized. The parallel constraint is given by:

$$\Omega_{\text{par}} = \sum_{\{i,j\} \in E'_{\text{metro}}} |s_{ig}(\theta_{ij} - \theta_{pq})|^2 \tag{3.8}$$

$$\text{where } s_{ig} = \min(1, \frac{1}{\|c_{ij} - d_i\| C_s}) \quad \text{and} \quad c_{ij} = v'_i + \frac{1}{2}(v'_j - v'_j)\ .$$

Where as c_{ij} is the center of the edge $\{i,j\} \in V'_{\text{metro}}$ and d_i is the closest point to c_{ij} on the closest edge $\{p,q\} \in E_{\text{guide}}$. θ_{ij} and θ_{pq} is the angle of the edge $\{i,j\} \in E_{\text{metro}}$ respectively $\{p,q\} \in E_{\text{guide}}$.

In case the difference between the angles θ_{ij} and θ_{pq} is $> \frac{\pi}{4}$, the edge $\{i,j\} \in E_{\text{metro}}$ should be normal to the guide edge $\{p,q\} \in E_{\text{guide}}$, because in cases as such, it is very unlikely that the edge can be rotated towards $\{p,q\}$ without distorting the metro map to much. This can lead to overlaps and very windy metro lines. Therefore Equation 3.8 is extended with

$$\Omega_{\text{par}} = \sum_{\{i,j\} \in E'_{\text{metro}}} |s_{ig} \min(|\theta_{ij} - \theta_{pq}|, |\theta_{ij} - \theta_{pq} + \frac{\pi}{2}|)|^2\ . \tag{3.9}$$

3.5.3 Edge Length Constraint

Modern metro maps commonly have uniform distances between stations, resulting in an elimination of information concerning distances between stations. But in general, it creates a clearer layout. In previous works [WC11] it is argued that the actual distance between stations is not of great importance, because the overall travel time depends more on the number of stations, rather than the specific distance. Ω_{len} is defined as:

$$\Omega_{\text{len}} = \sum_{\{i,j\} \in E'_{\text{metro}}} |(v'_i - v'_j) - s_{ij} R_{ij} (v_i - v_j)|^2, \tag{3.10}$$

$$\text{where} \quad s_{ij} = \frac{C_d}{|v_i - v_j|} \quad \text{and} \quad R_{ij} = \begin{bmatrix} \cos\theta_{ij} & -\sin\theta_{ij} \\ \sin\theta_{ij} & \cos\theta_{ij} \end{bmatrix}.$$

C_d is a constant influencing the target length of edges, and θ_{ij} is the unknown rotation angle of the edge $\{i,j\} \in E'_{\text{metro}}$.

3.5.4 Maximal Angles Constraint

In order to make it easier for users to track metro lines alongside the map, metro lines should be as straight as possible [Rob14]. Therefore the angle between incidents edges should be maximized. In case the degree of a vertex $v'_i \in V'_{\text{metro}}$ is 2, the inner angle of the two neighboring edges sharing v'_i should be π. Furthermore, if a vertex has a higher number of outgoing edges, the angles should be maximized to facilitate the distinction between different metro lines. This concept was described by Wang et al. first as

$$\Omega_{\text{max}} = \sum_{v'_i \in V'_{\text{metro}}} \sum_{\{\{i,j\},\{i,k\}\} \in N(i)} |v'_i - (v'_j + u'_{jk} + \tan(\frac{\pi - \theta_{ijk}}{2} u'_{jk})|^2 \tag{3.11}$$

$$\text{and } u'_{jk} = \frac{1}{2}(v'_k - v'_j) \ .$$

Where as θ_{ijk} is the calculated inner angle of the edges $\{i,j\}, \{i,k\} \in E'_{\text{metro}}$ and the set of all edges sharing v'_i is denoted by $N(i)$.

3.5.5 Positional Constraint

Metro maps are commonly schematic representations of a metro network, rather than a geographically accurate map, but the overall direction and the approximate relative location of a station should still be preserved. For example, if a metro station is located in the north of a city, the corresponding vertex that represents this station should be located in the top area of the map. If not, this could create a conflict with the image that users already have of the metro network and therefore could make navigation more difficult. Hence Ω_{pos} is added in order to preserve the initial location $v_i \in V_{\text{metro}}$. This constraint is defined as

$$\Omega_{\text{pos}} = \sum_{v'_i \in V'_{\text{metro}}} |v'_i - v_i|^2 \ . \tag{3.12}$$

3.5.6 Overlay Constraint

The constraints described in the previous sections do not ensure that a vertex $v'_i \in V'_{\text{metro}}$ does not overlap with another edge $\{j,k\} \in E'_{\text{metro}}$. In other words $\|d_i - v'_i\|$ is smaller than a defined minimum distance. Where as d_i is the geographically closest point on the closest edge $\{j,k\} \in E'_{\text{metro}}$ to $v'_i \in V'_{\text{metro}}$ with $j,k \neq i$. Preventing cases as such is crucial for the readability of the metro map, because otherwise overlaps could indicate a station along a metro line, which does not actually exist. Therefore the additional energy term Ω_{over} is defined. By minimizing

$$\Omega_{\text{over}} = \sum_{v'_i \in V'_{\text{metro}}} |(d_i - v'_i) - \max(1, \frac{\gamma}{\|d_i - v'_i\|})(d_i - v'_i)|^2 \ , \tag{3.13}$$

the overlaps are prevented. γ is the minimal allowed distance.

Wang et al. [WC11] described a similar constraint, where an energy term for a vertex is added if it violates the minimum distance. When the distance is sufficient the constraint is removed for this vertex. The constraint described and implemented in this thesis differs in that the energy term is not removed for the vertices, if these do not overlay with other edges. In Equation 3.13, $\Omega_{\text{over}} = 0$ in case no vertex is violating the minimum distance. Therefore it is not necessary to exclude the constraint for these vertices.

3.6 Mixed Layout Constraints

In this stage metro edges are rotated either octilinear (e.g. with an angle of 0, $\pm\frac{\pi}{4}$, $\pm\frac{\pi}{2}$, $\pm\frac{3\pi}{4}$, π) or towards the closest guiding edge. Metro edges $\{i,j\} \in \tilde{E}_{\text{metro}}$ which are rotated octilinear are referred to as octilinear edges. Edges that are rotated to be parallel to a section of the guide are called smooth edges. The constraints $\Omega_{\text{direction}}$ quantifies the differences between the angles of edges $\{i,j\} \in \tilde{E}_{\text{metro}}$ and their target angels. Furthermore, the two energy terms Ω_{over} and Ω_{close} are added. These constraints are equal to the one described in equation 3.13, respectively 3.6. In order to calculate the mixed layout the energy term Ω_{mixed} given by

$$\Omega_{\text{mixed}} = w_{\text{direction}}\Omega_{\text{direction}} + w_{\text{over}}\Omega_{\text{over}} + w_{\text{close}}\Omega_{\text{close}} \tag{3.14}$$

is minimized. The constraint $\Omega_{\text{direction}}$ is formulated as

$$\Omega_{\text{direction}} = \sum_{\{i,j\}\in E_{\text{metro}}} |(\tilde{v}_i - \tilde{v}_j) - R_{ij}(v'_j - v'_i)|^2 \ . \tag{3.15}$$

$$\text{Where } R_{ij} \text{ is given by } R_{ij} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

This constraint enforces edges to rotate by the angle θ. $\tilde{v}_i, \tilde{v}_j \in \tilde{V}_{\text{metro}}$ are the mixed positions of the metro vertices and $v'_i, v'_j \in V'_{\text{metro}}$ are the respective the smooth positions.

The primary goal of this project is to create a layout that includes the overall shape of the guide path and represents it as well as possible. The assumption is made that a section of the guiding path is represented well by a metro edge if it is parallel to the closest section of the guide and is not located to far away. Therefore the decision if an edge should be treated as octilinear or smooth is based on the angle of the metro edge and the distance to the closest section of the guide. These two concepts take into account that θ is defined as:

$$\theta = \begin{cases} \theta_{ij} - \theta_{\text{guide}}, & \text{if } \ |\theta_{\text{guide}} - \theta_{ij}| < \beta \wedge h_{ij} \\ \theta_{ij} - \theta_{\text{octi}}, & \text{otherwise} \end{cases} \tag{3.16}$$

$$\text{with } \beta = \frac{\pi}{8} \ ,$$

$$\text{and } h_{ij} = \min(\|v'_i - d'_i\|, \|v'_j - d'_j\|) < \|v'_j - v'_i\| C_m \ .$$

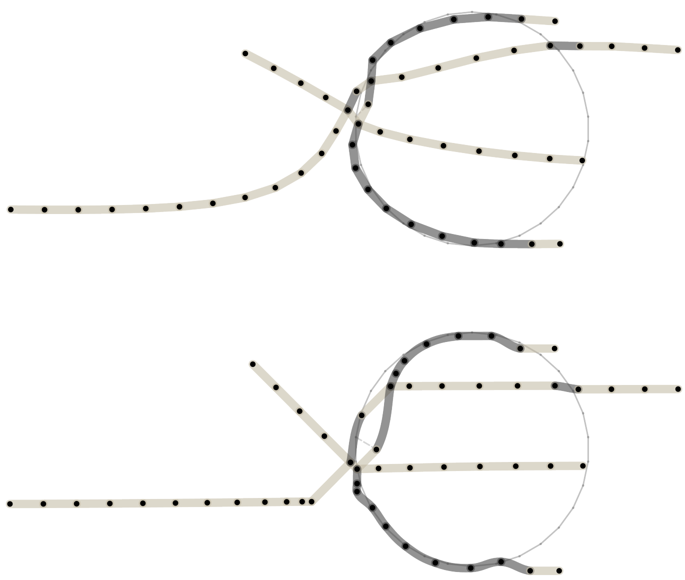

Figure 3.5: Smooth edges are displayed with a dark gray stroke. Octilinear are shown with a light gray stroke.

d'_i is the closest point to $v'_i \in V'_{\text{metro}}$ located on an guide edge $\{p, q\} \in E'_{\text{guide}}$. θ_{octi} is the closest octilinear direction. θ_{guide} is the angle of $(d_j - d_i)$ and C_m is a constant influencing how far away a smooth edge can be located. Figure 3.5 illustrates the distinction between octilinear and smooth edges.

3.7 Iterative minimization of Constraints

The optimization process aims to find the final vertex positions that minimize the constraints Ω as well as possible. It is achieved by iteratively calculating new positions until a sufficient result is found. A solution (e.g. the positions of the metro stations) is calculated by using a Conjugate Gradient method [HS+52]. The following section describes how the smooth positions V'_{metro} are calculated. The positions $\tilde{V}_{\text{metro}}$ and V'_{guide} are calculated in a same way, using the corresponding constraints and weights. V'_{metro} is calculated by satisfying the constraint Ω_{smooth}, which is reached by transforming the constraints Ω_{close}, Ω_{par}, Ω_{over}, Ω_{len}, Ω_{max} and Ω_{pos} and the unknown $v'_i \in V'_{\text{metro}}$ into a linear system

$$AV' = b(V') \tag{3.17}$$

that is calculated iteratively. Because the equation 3.17 is solved separately for the components x and y with $v'_i \in V'_{\text{metro}}$, V'_{metro} can be defined as $V'_{\text{metro}} = \left[v_{i.x}, v_{j,x}, \cdots, v_{i,y}, v_{j,y}\right]^T$. The matrix A is referred to as the Coefficient matrix and contains the weights of the corresponding constraints. For the smoothing step, the matrix A can be defined as

$$A_{\text{smooth}} = \begin{bmatrix} w_{\text{close}} & 0 & \cdots & 0 & 0 \\ 0 & w_{\text{close}} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & w_{\text{close}} & 0 \\ 0 & 0 & \cdots & 0 & w_{\text{close}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{\text{pos}} & 0 & \cdots & 0 & 0 \\ 0 & w_{\text{pos}} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & w_{\text{pos}} & 0 \\ 0 & 0 & \cdots & 0 & w_{\text{pos}} \end{bmatrix}.$$

In this case, the number of rows of A equal the number of constraints multiplied by $|V'_{\text{metro}}|$. The number of columns is the number of unknown in the system (e.g. $|V'_{\text{metro}}|$). Because the linear system given in equation 3.17 is over determined – there are more equations than unknown – new positions $v'_i \in V'$ can be calculated with

$$V' = (A^T A)^{-1} A^T b'. \tag{3.18}$$

$b(V')$ is a vector with the length of the number of unknown multiplied by the number of constraints. Each entry represents a valid solution for a single unknown regarding a single constraint. The Conjugate Gradient method [HS+52] should give a result after as many iterations as there are unknowns.

CHAPTER 4

Implementation

This chapter outlines the specifications of the implementation as well as the system used to test the approach. The interactive process for matching the metro stations with a guide vertex is described in section 4.2. Further, section 4.3 explains how a designer can choose which metro line is affected by the guide and section 4.4 describes how the calculated layout is visualized. section 4.5 outlines how a user can manipulate certain parameters and therefore further influence the layout.

4.1 System Specification

The algorithm was implemented with C++. QT5 was used to render the results as well as to create the user interface. To represent the metro networks the Boost Graph library was used. The Eigen 3 library was used to handle the large matrices needed for the optimization process. CMake 3.19 was used as the built environment. Examples in this thesis were created with a MacBook Pro 2011, equipped with an Intel Core i7 CPU and 4GB of RAM. Tests were made with an Ubuntu 20.19 system.

4.2 Interactive Matching

As outlined in the previous chapter, the resulting layout depends highly on the position of the guiding paths. Therefore an interactive process is implemented, where the designer can define the matching between guide vertices and metro stations by interacting with the graphic representation of the two graphs. This can be done by first clicking on a vertex of the guide graph and then on a metro stop. When the selection was successful the matching is indicated by a dotted line – you can see this process in Figure 3.4. The matching can then be removed by clicking either on the metro or the guide vertex. Each metro vertex can only be matched to one guide vertex and each guide vertex to a single metro stop. Generally it is not necessary to find a match for each guide vertex, because a small number of matches can already be quite sufficient.

The process of matching was implemented as a process where a human being is a part of the loop. By doing so contextual information about the city or the real-world metro network can be considered by the designer, because information as such is not encoded into the geographic positions or the topology of the metro networks, and therefore can not be taken into account by a system that uses an automated matching algorithm like Factorized Graph Matching [ZDlT12]. Furthermore, artistic considerations can also be taken into account with this process.

4.3 Effected Metro Lines

The user can define which metro lines should be affected by the guide. For the parts of the metro graph that do not take the guiding path into account, the constraints $\Omega_{\text{close}}, \Omega_{\text{par}}$ are removed when creating the smooth layout. Edges that are not affected by the guide will be treated as octilinear edges, regardless their positions and angles in relation to the guide path. Therefore these edges are rotated towards the closest octilinear angle.

4.4 Visualization

After each optimization process has found a solution, the resulting layout is displayed and can be exported as an SVG or PNG file. Metro edges are rendered in the corresponding color of the metro line. The color is defined in the source file of the metro network. Stations are displayed as dark dots. Because the complexity of different metro networks varies greatly, a uniform edge width for all metro networks is not suitable. By assuming that the complexity of a network mainly depends on the number of metro stations n. The line width Γ is calculated by $\Gamma = \frac{\epsilon\chi}{\sqrt{n}}$ with $\epsilon = 0.08$. Where χ is the window width, and ϵ can be varied depending on the desired appearance. This is a strong simplification, but the ideal line thickness depends on a variety of different considerations – for example being, how dense the center of the map compared to the outer parts is and for which purpose the map is designed. All in all, the line width calculated with the equation above produces good results for the majority of metro networks, that have been tested and is also the one used in the given examples in this thesis.

During the mixed stage a distinction between octilinear and smooth edges is made. Most of the tested guide paths approximate round or organic shapes. Therefore, metro edges which are treated as smooth edges and represent these guide paths are rendered as Bezièr curves. The start and end point of the curved edge $\{i, j\} \in \tilde{E}_{\text{metro}}$ is equal to

$\tilde{v}_i, \tilde{v}_j \in \tilde{V}_{\text{metro}}$. The two control points of the Bezièr curve t_1, t_2 are given by:

$$
\begin{aligned}
t_1 &= \frac{t_{1,2} - t_{1,1}}{2} + t_{1,1} \ , \\
\text{with } t_{1,1} &= \tilde{v}_i + \frac{1}{4} \frac{\|\tilde{v}_i - \tilde{v}_j\|}{\|\tilde{v}_i - \tilde{v}_k\|} (\tilde{v}_i - \tilde{v}_k) \ , \\
\text{and } t_{1,2} &= \tilde{v}_i + \frac{1}{4} (\tilde{v}_j - \tilde{v}_i) \ .
\end{aligned}
\tag{4.1}
$$

Where $\{i, j\}, \{k, i\} \in \tilde{E}_{\text{metro}}$ are two neighboring edges and $\{i, j\}$ are treated as smooth edges. t_2 can be calculated similarly by taking the two edges $\{i, j\}, \{j, m\} \in \tilde{V}_{\text{metro}}$ into account.

4.5 Parameters

The method presented in this thesis proposals an interactive process where the user has more control over the resulting map, compared to previous works [WC11, OMI18, NW10].

For doing so a graphic interface was implemented where the designer of the metro map can manipulate multiple variables and weights. A screenshot of the interface is shown in Figure 4.1. These parameters affect the resulting layout of the algorithm. In case the resulted metro map does not meet the desired requirements, the user can recalculate the last step with different variables. The user can also change the weights w described in equation 3.5 and 3.14 without recompiling or restarting the program. Furthermore, the minimum allowed distance between a metro vertex and a metro edge – denoted by γ in equation 3.13 – as well as the thickness of the metro edge can be manipulated.

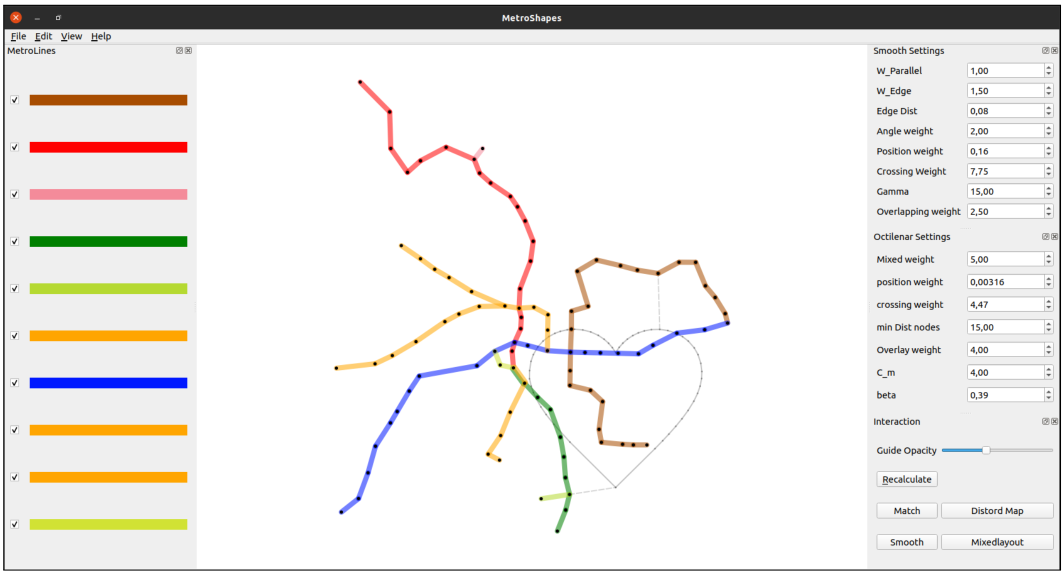

Figure 4.1: Screenshot of the interface.

CHAPTER 5

Results & Discussions

In this chapter, some of the different results are presented and discussed regarding the design objectives outlined in section 3.1. The effectiveness and impact of the guide paths on the resulting layout are demonstrated in section 5.3. In order to visualize that two maps were created for multiple cities: one with and one without a guide path. Additionally, several examples of the Vienna metro map were generated showing how the map behaves when using guide paths of different shapes. In section 5.4 the limitations of the suggested process are discussed.

5.1 Test-cases

Metro Network	Number of Stations	Number of Edges
Single Line	23	18
Lyon	40	40
Prague	30	30
Vienna	90	96
Taipei	96	98
Berlin	170	182
Moscow	199	242
Paris	236	268

Figure 5.1: The number of metro stations and edges between stations for each tested metro network.

In order to evaluate the developed method numerous metro networks have been tested – ranging from small synthetic networks with 23 stations to huge metro networks like in Paris with 236 stations and 268 edges.

Guide	Number of Vertices	Number of Edges
Straight Line	2	1
Curve	15	14
Sin wave	19	18
Spiral	43	42
Heart	53	53
Circle	54	54
Cloud	76	76
Concentric circles	110	110
Eye	118	118

Figure 5.2: The number of vertices and edges for each guide.

The guide paths used in the examples given were created with a vector drawing program and a custom python script that converts the paths that were drawn to the needed format.

Some of these guides are quite complex – for example the spiral shape – and others are relatively simple consisting of only a single edge. Figure 5.1 lists all of the tested metro maps including the number of stations and edges of each network. The number of vertices of each guide is listed in Figure 5.2.

5.2 Results

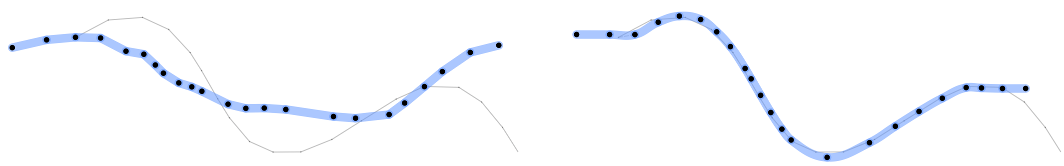

Figure 5.3: Single Metro Line with the sin wave guide. Left: initial layout. Right: resulting layout.

Several metro networks were tested using different guides – ranging from simple lines to complex shapes. The requirements of the metro maps generated with this approach can be summarized in two characteristics:

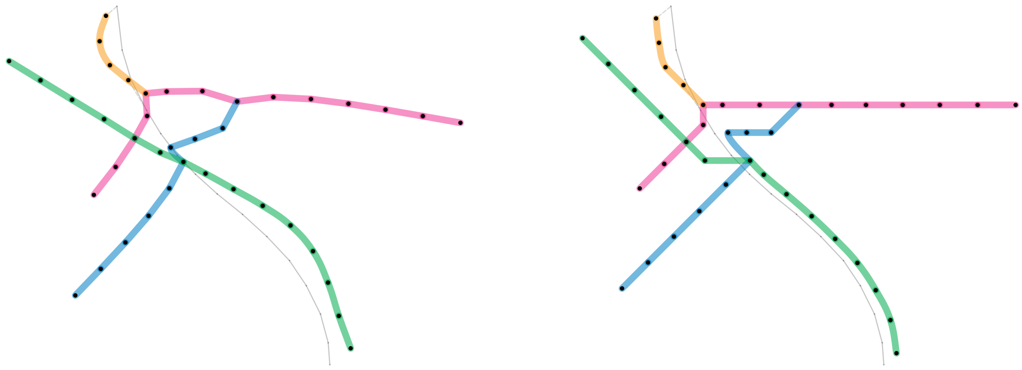

Figure 5.4: Lyon Metro Map with curve guide. Initial and resulting layout.

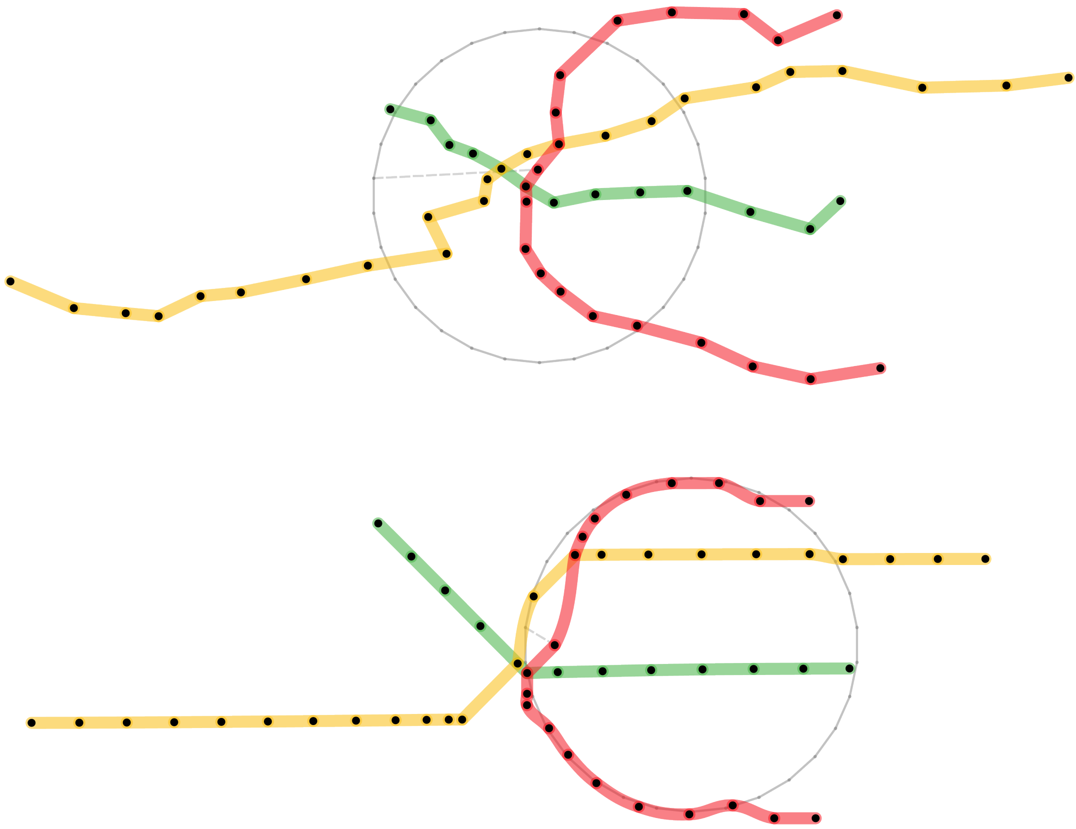

Figure 5.5: Prague Metro Map with circle guide. Initial and resulting layout.

- The maps should full fill the design criteria of a modern metro map.
- The metro map should be influenced by the guide path. It should replicate and include the shape as well as possible.

A detailed elaboration of the design criteria can be found in section 3.1.

Some of the design criteria are conflicting each other. An example being, that the relative positions of two metro stations should be preserved in the resulting map, as well as, that the guide paths should be replicated as precisely as possible. Therefore, the weight of the constraints has to be chosen according to the use case of the map and the designer's preferences. It could be argued that for a metro map that primarily has an artistic purpose the preservation of the relative position is less crucial than the replication of the guide shape. In contrast to that being a map that is primarily used by locals for navigating from one station to another. In this case usability concerns would potentially outperform a more visually pleasing and interesting layout.

Testcase	Transforming (s)	Smooth (s)	Mixed (s)	Overall (s)
Single Line Figure 5.3	< 0.01	0.99	0.04	0.15
Lyon Metro Figure 5.4	< 0.01	0.66	0.29	0.96
Prague Metro Figure 5.5	0.01	2.06	0.91	2.99
Taipei Figure 5.6	0.01	18.81	8.34	27.164
Berlin Figure 5.11	0.02	180.75	78.54	259.3
Moscow Figure 5.12	0.1	366.03	153.47	519.6
Paris Hearth Figure 5.8	0.01	682.47	292.84	975.32
Paris Eye Figure 5.9	0.37	690.46	290.54	981.38
Paris Cloud Figure 5.10	0.11	685.57	291.55	977.242

Table 5.1: Measured times to compute the different layouts. All values are in seconds.

The amount of time needed to compute the resulting layout is listed in Table 5.1. Further, the Table also lists the time needed to translate the guide, generate the smooth layout as well as calculate the mixed layout. The tests were conducted on a rather old machine, described in section 4.1.

In the first example given the metro network consists of only a single line that is then transformed – this can be seen in Figure 5.3. As you can see the metro line follows the guide path very well in the resulting layout. Stations that were initially located not too far away from the guide path moved very close to the guide path. The majority of the metro edges were rotated in order to run parallel to the guide. Only the edges of metro line ends were rotated in an octilinear direction. In the resulting map the stations are spaced out relatively evenly and the metro line has no unwanted twists.

The two smaller real-world metro networks that were tested are the ones in Lyon and Prague. The resulting layouts can bee seen in Figure 5.4 and 5.5. Tests of the metro network in Lyon were done with a guide having the shape of a simple curve and the network in Prague with a circular-shaped guide. The network in Lyon integrates the shape of the guide very well into the resulting layout, but Prague's metro network only portrays the left side of the shape – in this case a circle – very well. No metro lines follow the guide path on the right side of the shape. This is, because the stations are only transformed based on local constraints – the stations are moved towards the closest part of the guide shape. If the guide is placed in a way that some parts of the guide are to far away from a metro line these parts of the guide shape can not be replicated by a metro line and therefore these sections are also not reproduced in the resulting layout. In both cases there are no overlaps and the distance between two stations is bigger than the critical minimum. Only in the example of Prague the stations located in the center of the city and therefore also map are a bit clustered and therefore making navigation in this area potentially challenging.

Tests of the metro network in Taipei were made with a guide in the shape of a heart. The result can be seen in Figure 5.6. The generated layout has no overlaps, stations are spaced out evenly and sections of the metro network that do not contribute to the recognition of the heart shape are not unnecessarily twisted. The majority of the metro edges are rotated to run parallel to a section of the guide. An example being here the stations of the yellow line on the top right which are then placed so that they contribute to the recognition of the heart in the resulting layout, although these are not located directly along the guide path.

Paris has a huge metro network consisting of 16 metro lines and 236 stations. Therefore, potentially more complex guide shapes could be represented in a network as such. Three layouts were generated for Paris that are more experimental having more of artistic purpose in mind. The layout influenced by the shape of a heart is presented in Figure 5.8 and 5.7. An example with the guide shaped as a cloud can be seen in Figure 5.10 and in Figure 5.9 a layout is shown that is affected by a guide shaped as an abstract illustration of an eye. The map created with the heart-shaped guide represents the shape of the guide with the network quite well and the heart could even be recognized if the guide would not be displayed – as you can see in Figure 5.7. This is not the case for the other layouts of Paris presented, where the shape of the guide is hard to recognise without visual support. This could possibly also be because these shapes – the cloud an the eye – are more complex and at the same time also less iconic compared to the shape of a heart. In the three examples given, the metro lines are influenced by the guide shape. Only a small number of overlaps, short edges or unnecessary twists can be seen. In the author's view, these three examples generate a visually interesting results – the heart shaped network being the one representing the shape very accurately.

Figure 5.11 shows the metro map of Berlin that is here influenced by a rather complex guide in the shape of a spiral. This example demonstrates the limitations of the presented process. The metro lines and positions of the stations are influenced by the guide shape. The purple line for example follows the guide shape quite well, but the over all resulting layout neither represents the guide shape well nor does the map make navigation easy.

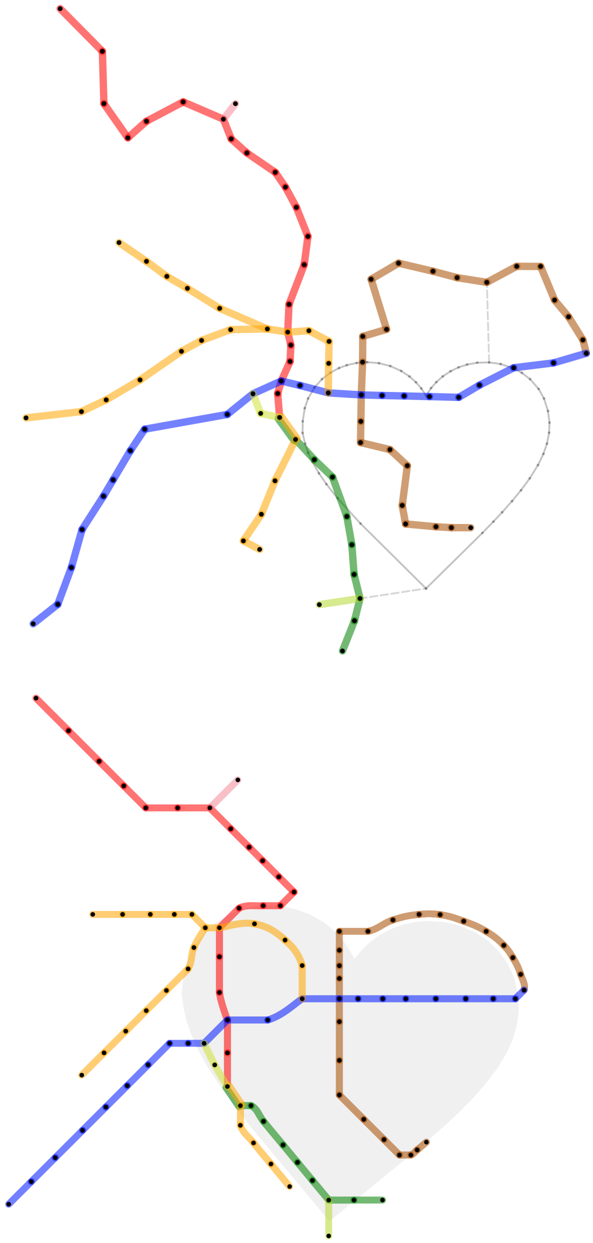

Figure 5.6: Taipei Metro Map with heart guide. For a better visibility, the hearth is colored gray in the resulting layout.

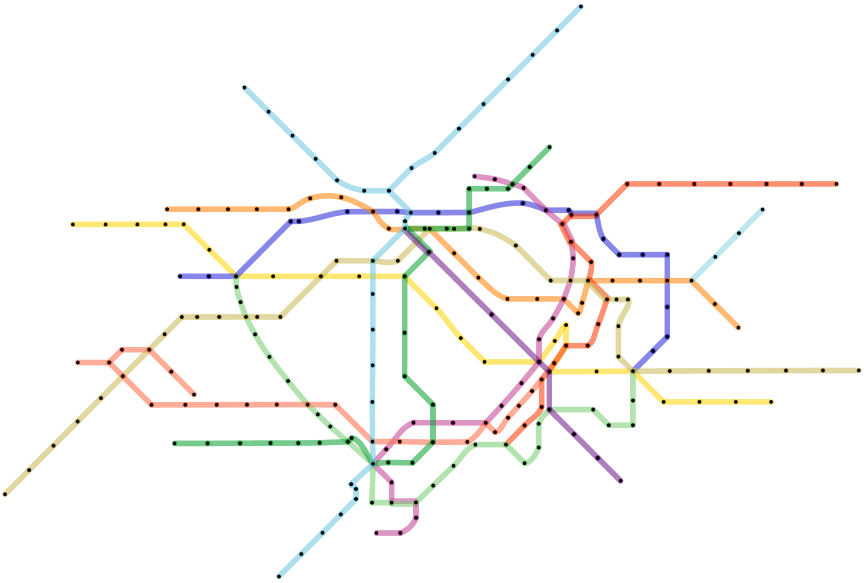

Figure 5.7: Paris metro map with heart guide.

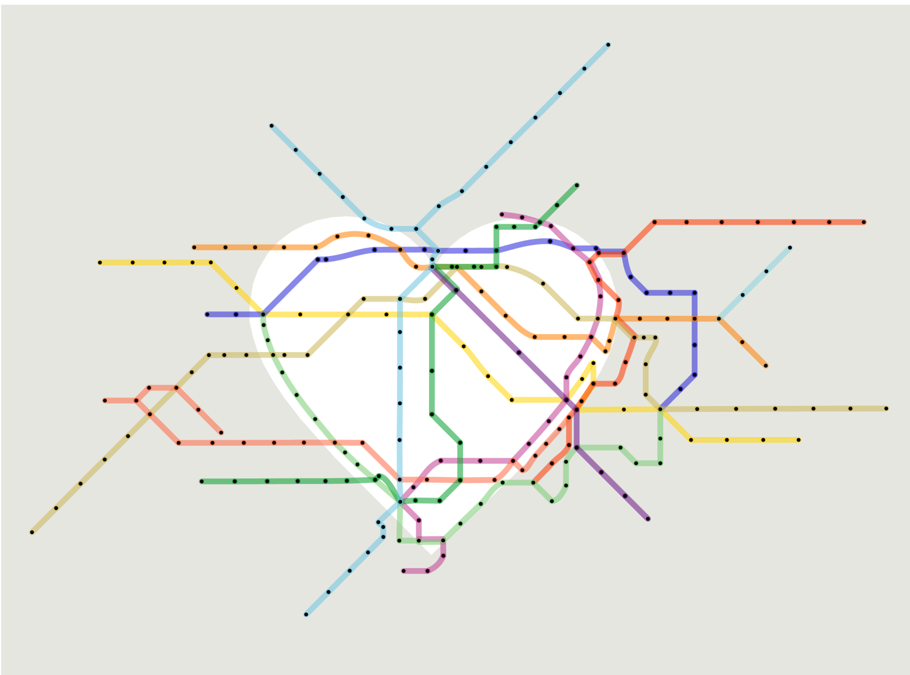

Figure 5.8: Paris metro map with heart guide. The guide has been colored afterward.

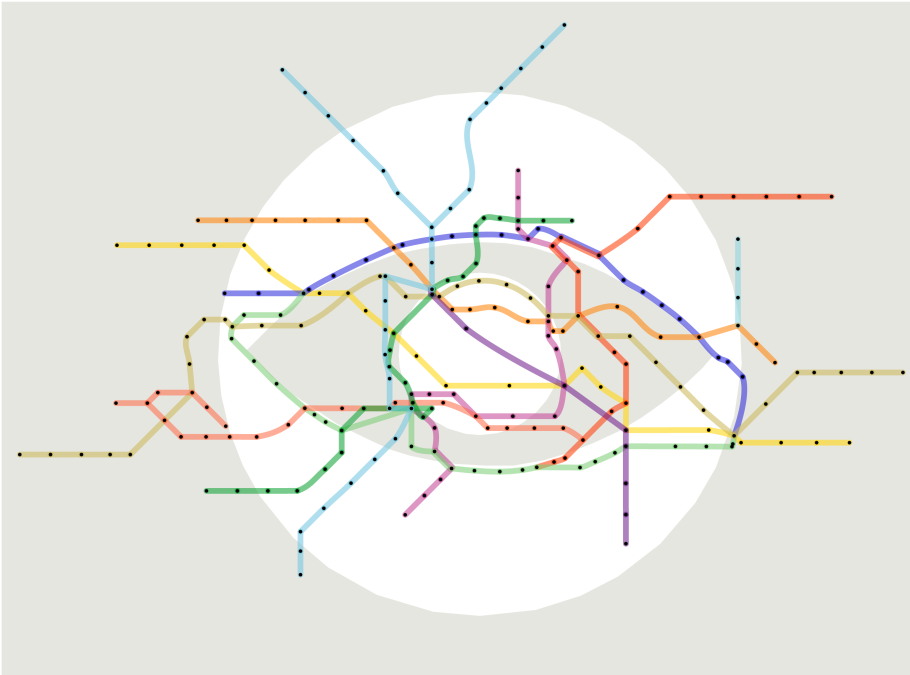

Figure 5.9: Paris metro map with eye guide. The guide has been colored afterward.

Figure 5.10: Paris metro map with cloud guide. The guide has been colored afterward.

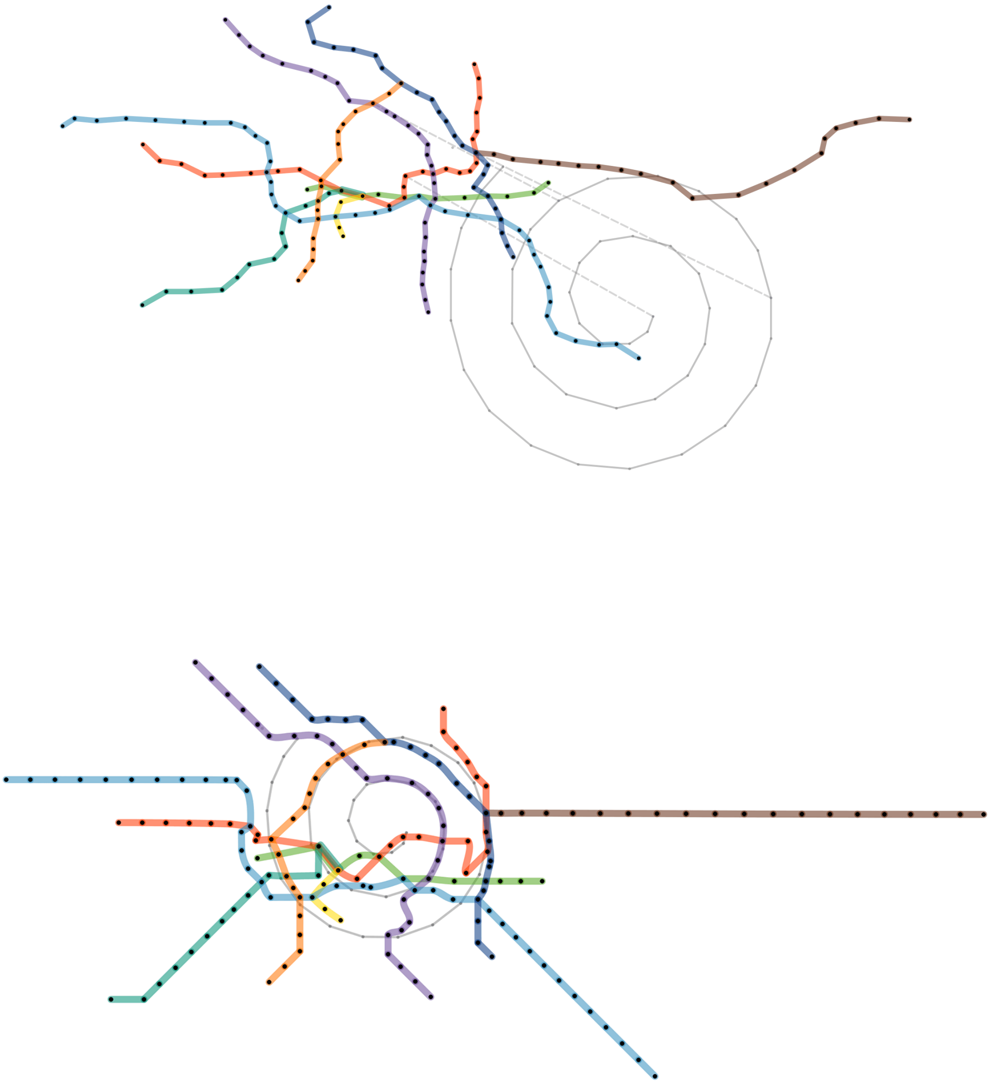

Figure 5.11: Berlin Metro Map with spiral guide

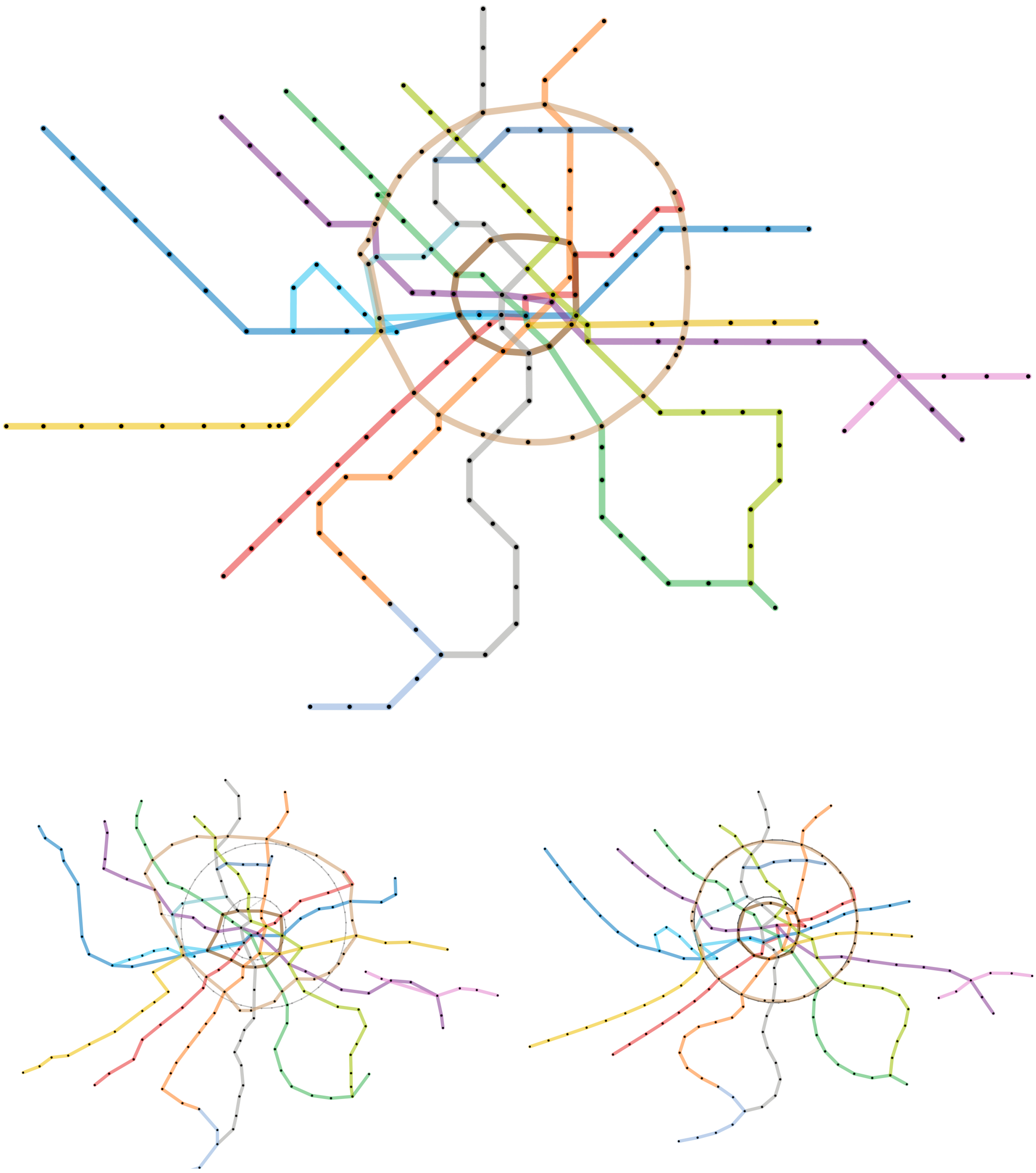

Figure 5.12: Moscow Metro Map with two concentric circles. Top: resulting layout; Left: layout with the transformed guide; Right: smooth layout.

In all the examples given the shape of the guide affects all of the metro lines in the given network. In Figure 5.12 an example of the Moscow metro network can be seen, where the guide only effects two lines (the ones in brown). In the geographically accurate representation of the network these metro lines shape two irregular circles. By applying two circular guide paths these lines were transformed into two concentric circles. The result is similar to handmade metro maps of Moscow that are shown in Figure 2.1. The circles are well represented and recognizable in the resulting layout, but the center of the map is very dense, which makes it difficult to distinguish the metro lines from one another.

5.3 Impact of the Guide Path

To demonstrate what effect the guide paths have on the resulting layout the maps are once generated without and then with guide paths. Both versions use the same algorithm and weights. The results can be seen in Figure 5.13. On the left side you can always see the octilinear layout that was generated without a guide and next to it on the right are the mixed layouts which were generated by adding guide paths. As you can see in the examples given the versions differ greatly from one another. Naturally, the octilinear layout produces metro maps that full fill the design criteria that generally apply better. But using layouts with guide paths gives designers the chance to influence the metro maps to a bigger extent and potentially also create symbolic shapes, that make navigation through a cities or the understanding of the map in itself easier.

In Figure 5.14 four different versions of the Vienna metro map are shown. In order to illustrate the effect that the guide paths have, four different guide shapes were used for influencing the layout of the metro network.

5.4 Limitations

Since the metro lines are only transformed based on local constraints, this naturally comes with numerous limitations. The final layout is highly dependant on the relative position of the guide to the metro network – respectively on the matching between guide vertices and metro stations. Even if the guide path is placed well, there are combinations of metro networks and guide paths, where the resulting layout is not sufficient – an example being the Berlin metro map with the spiral guide as shown in Figure 5.11. Where neither the guide is represented well nor the resulting layout makes navigation easy. Further, if a guide path is placed to close to a dense center, numerous metro lines are strongly affected by the same guide sections, which potentially leads to unclear layouts. In most cases it was possible to prevent overlaps between metro stations and edges that were not connected, but the more complex the network was, the more common it was to have overlaps. The design criteria defined in section 3.1 are partly contrary. Therefore a designer using this method has to choose the weights for the corresponding constraints wisely. It is about finding a balance between the constraints aiming to make a metro

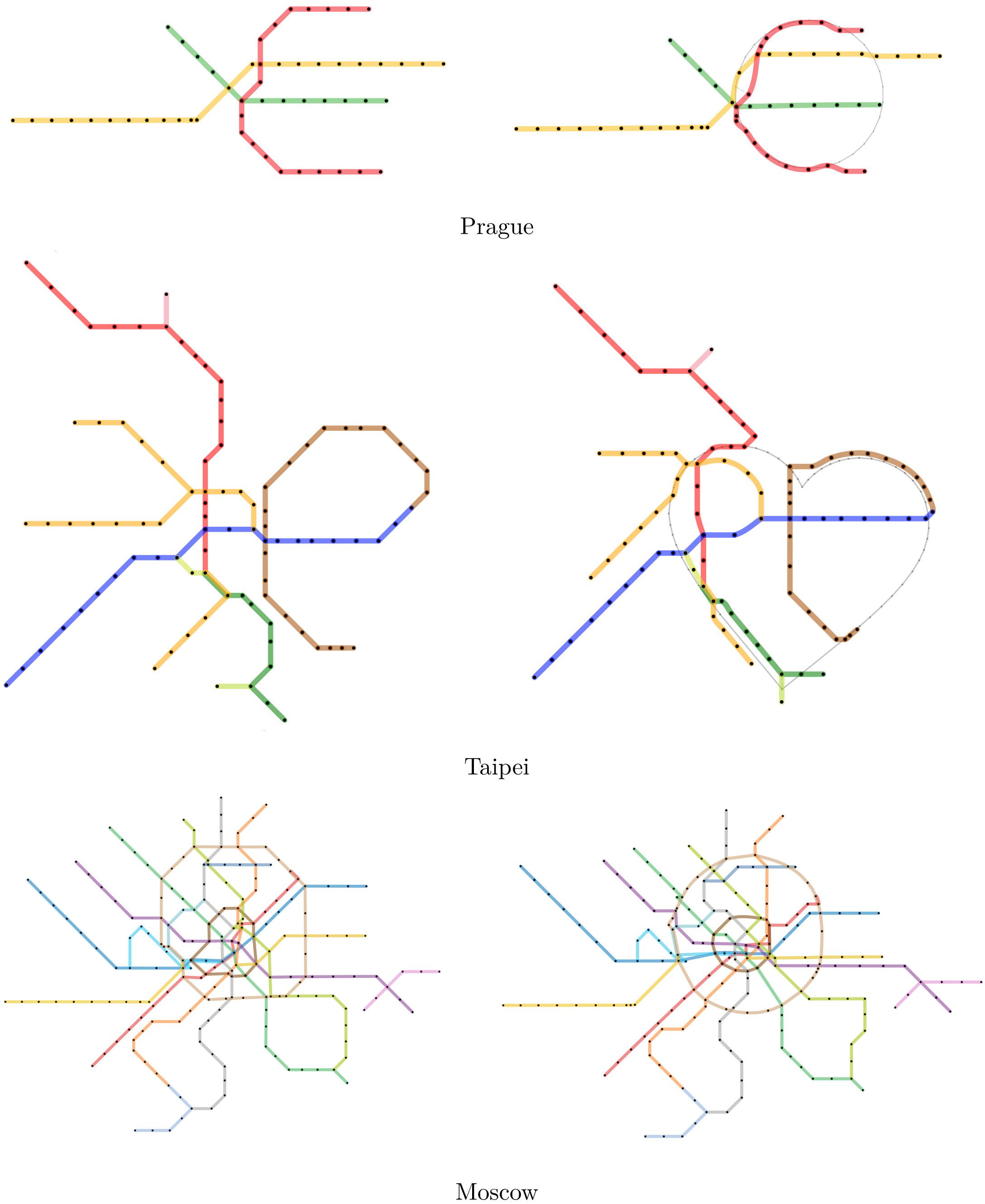

Figure 5.13: Effects of guide paths. Left column: metro maps generated without guide path. Right column: mixed layout created with guide edges.

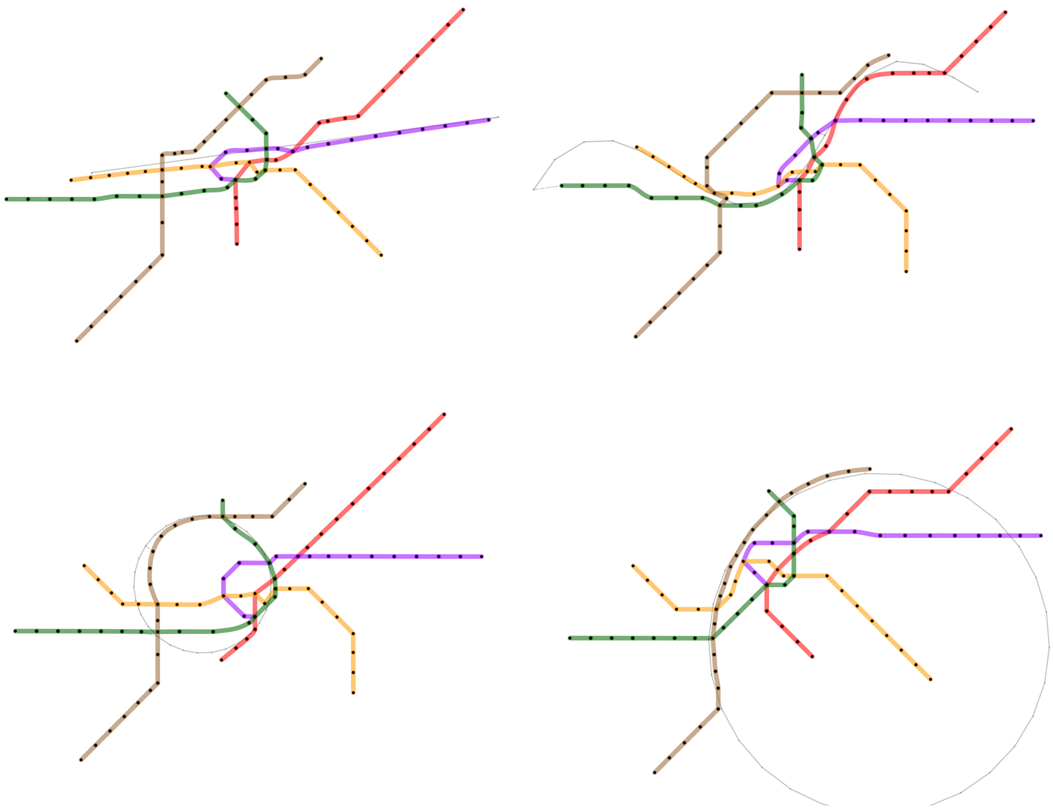

Figure 5.14: Vienna metro network with different guides. Top left: straight line; top right: curve guide; bottom left: small circle; bottom right: large circle.

map better for navigation and capturing the overall shape of a guide path well with the metro network itself. Furthermore, the solution found by minimizing the constraints iteratively describes only a local minimum and potentially better solutions could exist than the one found.

CHAPTER 6

Conclusion

In the last chapter a short summary is given and potential future work is outlined.

6.1 Summary

In most state-of-the-art metro layout algorithms there are few possibilities for designers to influence the outcome of a metro map. Resulting in not being able to take human knowledge as well as contextual information of a city into account when designing maps with these algorithms. In this thesis, an approach is presented where designers can influence the resulting layout based on a set of guide paths. The generated maps use a mix of two graph layout types. Therefore parts of the map – ones that are positioned close to a guide path – are transformed so that they run parallel to the guide and other parts run octilinear. The presented results have their weaknesses and the outlined method has it's downsides, but with the use of guide paths it is possible to generate visually pleasing metro maps. With this approach one can also generate maps that display parts of the metro line in symbolic shapes. An example being the metro system in Moscow, where two metro lines shape concentric circles, similar to the official Moscow metro map that is currently also used. The method outlined in this thesis can be used to generate layouts for smaller metro networks as well as larger networks consisting of more than 200 stations.

6.2 Future Work

The presented approach has several limitations, which could be improved in future works. One of the examples being, that there are certain combinations of metro networks and guide paths, where some stations are placed to close to one another and therefore can not be kept apart in the resulting layout. It would also be interesting to develop a version which enables real-time metro map and guide path editing, because currently guide paths can not be drawn and edited in the developed framework and therefore makes the process relatively time-consuming. Further, it would be worthwhile adapting the algorithm as well as the constraints so that users could define which parts of a metro line should and which parts should not be affected by the guide path. It could also be interesting to further investigate into which visual features are of importance when recognizing specific shapes – as human beings – when these are incorporated into metro networks or any other graphs.

List of Figures

List of Tables

Bibliography

[CR14] Daniel Chivers and Peter Rodgers. Octilinear force-directed layout with mental map preservation for schematic diagrams. In *International Conference on Theory and Application of Diagrams*, pages 1–8. Springer, 2014.

[FHN+12] Martin Fink, Herman Haverkort, Martin Nöllenburg, Maxwell Roberts, Julian Schuhmann, and Alexander Wolff. Drawing metro maps using bézier curves. In *International Symposium on Graph Drawing*, pages 463–474. Springer, 2012.

[HMdN06] Seok-Hee Hong, Damian Merrick, and Hugo AD do Nascimento. Automatic visualisation of metro maps. *Journal of Visual Languages & Computing*, 17(3):203–224, 2006.

[HS+52] Magnus Rudolph Hestenes, Eduard Stiefel, et al. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS Washington, DC, 1952.

[LL00] Longin Jan Latecki and Rolf Lakamper. Shape similarity measure based on correspondence of visual parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1185–1190, 2000.

[LLHL14] Shih-Syun Lin, Chao-Hung Lin, Yan-Jhang Hu, and Tong-Yee Lee. Drawing road networks with mental maps. *IEEE transactions on visualization and computer graphics*, 20(9):1241–1252, 2014.

[Met19] Moskovsky Metropoliten. Moscow metro map & central circle, 2019. `https://mosmetro.ru/mcc/moscow-central-circle/`, accessed 2021-03-08.

[NN20] Soeren Nickel and Martin Nöllenburg. Towards data-driven multilinear metro maps. In *International Conference on Theory and Application of Diagrams*, pages 153–161. Springer, 2020.

[Nöl14] Martin Nöllenburg. A survey on automated metro map layout methods. In *Schematic Mapping Workshop*, 2014.

[NRW19] Benjamin Niedermann, Ignaz Rutter, and Matthias Wolf. Efficient algorithms for ortho-radial graph drawing. *arXiv preprint arXiv:1903.05048*, 2019.

[NW10] Martin Nollenburg and Alexander Wolff. Drawing and labeling high-quality metro maps by mixed-integer programming. *IEEE transactions on visualization and computer graphics*, 17(5):626–641, 2010.

[OMI18] Masahiro Onda, Masaki Moriguchi, and Keiko Imai. Automatic drawing for tokyo metro map. In *European Workshop on Computational Geometry (EuroCG'18)*, page 62, 2018.

[Ove15] Mark Ovenden. *Transit maps of the world*. Penguin, 2015.

[PCZ+20] Jiacheng Pan, Wei Chen, Xiaodong Zhao, Shuyue Zhou, Wei Zeng, Minfeng Zhu, Jian Chen, Siwei Fu, and Yingcai Wu. Exemplar-based layout fine-tuning for node-link diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[RGL17] Maxwell J Roberts, Hannah Gray, and Jennifer Lesnik. Preference versus performance: Investigating the dissociation between objective measures and subjective ratings of usability for schematic metro maps and intuitive theories of design. *International Journal of Human-Computer Studies*, 98:109–128, 2017.

[RNL+13] Maxwell J Roberts, Elizabeth J Newton, Fabio D Lagattolla, Simon Hughes, and Megan C Hasler. Objective versus subjective measures of paris metro map usability: Investigating traditional octolinear versus all-curves schematics. *International Journal of Human-Computer Studies*, 71(3):363–386, 2013.

[Rob14] Maxwell J Roberts. What? s your theory of effective schematic map design? 2014.

[RV16] Maxwell J Roberts and CN Vaeng. Expectations and prejudices usurp judgements of schematic map effectiveness. 2016.

[SR05] Jonathan M Stott and Peter Rodgers. Automatic metro map design techniques. In *Proceedings of the 22nd International Cartographic Conference*, 2005.

[SRMOW10] Jonathan Stott, Peter Rodgers, Juan Carlos Martinez-Ovando, and Stephen G Walker. Automatic metro map layout using multicriteria optimization. *IEEE Transactions on Visualization and Computer Graphics*, 17(1):101–114, 2010.

[WC11] Yu-Shuen Wang and Ming-Te Chi. Focus+ context metro maps. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2528–2535, 2011.

[WNT+20] Hsiang-Yun Wu, Benjamin Niedermann, Shigeo Takahashi, Maxwell J Roberts, and Martin Nöllenburg. A survey on transit map layout–from design, machine, and human perspectives. In *Computer Graphics Forum*, volume 39, pages 619–646. Wiley Online Library, 2020.

[WP15] Yu-Shuen Wang and Wan-Yu Peng. Interactive metro map editing. *IEEE transactions on visualization and computer graphics*, 22(2):1115–1126, 2015.

[ZDlT12] Feng Zhou and Fernando De la Torre. Factorized graph matching. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 127–134. IEEE, 2012.