FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Discrete Optimization on Graphs and Grids for the Creation of Navigational and Artistic Imagery

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Dipl.-Ing. Michael Birsak, BSc
Matrikelnummer 0525386

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Associate Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer
Zweitbetreuung: Dipl.-Mediensys.wiss. Dr.techn. Przemyslaw Musialski

Diese Dissertation haben begutachtet:

Peter Wonka	Robert Sablatnig

Wien, 30. August 2018

Michael Birsak

FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Discrete Optimization on Graphs and Grids for the Creation of Navigational and Artistic Imagery

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Dipl.-Ing. Michael Birsak, BSc
Registration Number 0525386

to the Faculty of Informatics

at the TU Wien

Advisor: Associate Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer
Second advisor: Dipl.-Mediensys.wiss. Dr.techn. Przemyslaw Musialski

The dissertation has been reviewed by:

Peter Wonka

Robert Sablatnig

Vienna, 30th August, 2018

Michael Birsak

Erklärung zur Verfassung der Arbeit

Dipl.-Ing. Michael Birsak, BSc
Haberlandtgasse 24/4/13
1220 Wien

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, 30. August 2018

Michael Birsak

Acknowledgements

First, I want to thank my PhD supervisor Prof. Michael Wimmer, who gave me the opportunity to work as a research assistant at the Institute of Computer Graphics after I finished my master studies for all the inspiring discussions and professional advice. Next, I would like to express my deep gratitude to my second advisor Przemyslaw Musialski for all the support and excellent collaboration. It was an honor for me to work with him on my projects and to sacrifice weekends to reach some deadlines. Great thanks also go to Prof. Peter Wonka for many helpful discussions and great ideas that guided me through difficult phases of my research work.

I also want to thank Meister Eduard Gröller for all the anecdotes about Goethe, Wittgenstein and many other writers that he performed during the KVs and social events with which he did not only amuse me and my colleagues but with which he could also broaden my horizon. A big Thank You goes to the head of the institute, Prof. Werner Purgathofer, for the way he was leading the group and especially for his great advice for improving all my talks. It was further an honor for me to be part of his relay team for the Vienna City Marathon (VCM).

Many thanks go to Andreas Weiner, Stephan Bösch-Plepelits, and Simone Risslegger, the always-on technicians of the institute, for the immediate and always friendly technical support. I particularly want to thank Stephan for his kind introduction to and always answering my questions about the OpenStreetMap. Let me also thank Anita Mayerhofer, Tammy Ringhofer and Max Höfferer, the secretaries of the institute, for their incredible support and always having an open ear for all kinds of issues.

I would like to thank Florian Rist for the collaboration in my last project, his way of thinking close to the limits of possibility and his catching euphoria during discussions about new and complex ideas.

A special thanks goes to my parents for their invaluable support during my studies. Most of all I want to thank my beloved Caroline and our daughter Lucia Josephine for their emotional and motivational support during all the hard times. I couldn't have made it without you.

This work was funded by the Austrian Science Fund (FWF), project numbers P23237-N23 and P27972-N31 and the Vienna Science and Technology Fund (WWTF ICT15-082).

Kurzfassung

Optimierung hat sich zu einem wichtigen Werkzeug in der Wissenschaft entwickelt und wird zur Auffindung einer Lösung bei vielerlei Problemen eingesetzt. Ein wichtiges Teilgebiet das besondere Beachtung findet ist *Diskrete Optimierung* welche sich durch ganzzahlige Variablen in der Problemformulierung auszeichnet. Die spezielle Aufmerksamkeit, die Diskrete Optimierung bekommt, ist durch den Zusammenhang zu bekannten kombinatorischen Problemen, welchen man oft begegnet, sowie durch Probleme welche sich mit keiner bekannten Methode effizient lösen lassen begründet. Zwei Domänen, die aufgrund ihrer inherenten diskreten Struktur auf fast natürliche Art und Weise eine Vielzahl von diskreten Optimierungsproblemen offenbaren sind *Graphen* und *Grids*.

In dieser Dissertation werden drei Publikationen präsentiert, welche in zwei Themengebiete eingeordnet werden können – Navigation und Kunst – und welche zwei Aspekte gemeinsam haben: sie basieren auf diskreter Optimierung und verwenden Graphen und Grids als zentrales Rechenmodell. Zunächst präsentieren wir eine Methode zur *Automatischen Generierung von Touristenbroschüren*. Diese Broschüren sind statische Karten welche sowohl wichtige Informationen über Sehenswürdigkeiten, als auch Instruktionen zur Navigation zwischen beliebigen Paaren von Sehenswürdigkeiten beinhalten. Wir beziehen mehrere anerkannte Design-Richtlinien in unseren Optimierungsansatz mit ein, um einfach lesbare Karten zu erhalten, welche anschließend auf Papier gedruckt werden können.

Als Nächstes erzielen wir einen Wechsel von statischen zu dynamischen Ergebnissen. Wir stellen ein System zur *Dynamischen Pfaderkundung auf Mobilen Geräten* vor, welches dem Anwender die Erzeugung von Visualisierungen zur einfachen Erkundung von unbekannten Regionen erlaubt. Wir setzen erneut Diskrete Optimierung ein, um einen interessanten Pfad sowie Positionen für rechteckige Objekte, welche wichtige Information über die Sehenwürdigkeiten in der Umgebung beinhalten, zu berechnen.

Zu guter letzt wenden wir unsere Erkenntnisse über Diskrete Optimierung an um künstlerische Ergebnisse zu bekommen und stellen einen vollautomatischen Fabrikationsweg zu Erzeugung von *String Art* vor. Wir verwenden unseren vorgestellten Algorithmus um eine Sequenz von Fadenabschnitten zu erhalten welche die Fabrikation mit einem zusammenhängenden Stück textilem Faden ermöglicht. Für die eigentliche Fabrikation stellen wir ein Hardware-Setup vor, welches einen hochpräzisen Industrieroboter, einen Fadenspanner einer professionellen Strickmaschine sowie ein selbstentwickeltes Wickler-Werkzeug beinhaltet.

Abstract

Optimization has become an important tool for many research communities and is used to find a good feasible solution for a variety of problems. One important subarea that attracts special interest is *discrete optimization*, which is characterized by integer variables in the problem formulation. The particular attention discrete optimization gets is caused by the close relation to some well-known combinatorial problems encountered in real life as well as hard problems for which methods that can solve them in polynomial time are unknown. Due to their inherent discrete structure, two domains that almost naturally reveal a multitude of discrete optimization problems are *graphs* and *grids*.

In this thesis, three peer-reviewed publications are presented that can be categorized into two different topics – navigation and arts – that have one fundamental aspect in common: they are all based on discrete optimization and use graphs and grids as their central computational domains. First, we present a method for the *Automatic Generation of Tourist Brochures*. These brochures are static printable maps that provide both essential information about some points of interest (POIs) as well as navigational instructions to tell the user how to travel between any pair of POIs. We incorporate several approved design guidelines into our optimization approach to obtain maps that are easily readable and can be printed on paper without visual clutter.

Second, we try to improve on the findings from our first publication to accomplish a switch from a static to a dynamic output. We propose a framework for *Dynamic Path Exploration on Mobile Devices* that allows the user to create custom visualizations that allow an easy exploration of an unfamiliar region. Again, we utilize discrete optimization to compute an interesting path through the environment as well as to find positions for rectangular entities that encapsulate important information about the POIs in proximity to the path.

Finally, we utilize our insights about discrete optimization to obtain an artistic output. In our proposed system, we aim at a fully automatic fabrication pipeline to create *String Art*. We present a greedy solver for binary non-linear least squares problems together with an adaption of an algorithm used for Euler-Path-Computation to obtain a sequence of strings that allows the fabrication using one continuous piece of textile thread. For the actual fabrication, we propose a hardware setup using a high-precision industrial robot, a tension regulator from a professional knitting machine and a custom-made winding tool.

Contents

CHAPTER 1

Introduction

Optimization has become a fundamental part for a majority of publications in the computer graphics field. While the strategies, runtimes and the memory consumptions of the approaches may differ essentially, they usually have a common purpose: For a given problem, they aim at finding the best solution amongst a possibly big set of feasible solutions. A solution is called *feasible* if it fulfills a set of given linear or even non-linear constraints. In the easiest case, the best solution is identified by mapping all feasible solutions to a real number that indicates their quality and finding the one solution that maximizes (or minimizes) this number. The mapping is performed by a special function that takes a solution as input and provides a real number as output and is usually referred to as the *objective* [BV04].

1.1 Discrete Optimization

While optimization in general deals with variables of a multitude of numeric data types, discrete optimization forces the variables to be strictly discrete and therefore limits the data types to be integer or binary. Discrete optimization problems are often encountered in reality and contain many of the famous problems for which efficient algorithms that can solve them in polynomial time are unknown. Such problems are therefore said to be *NP-hard* to express their non-polynomial complexity.

A famous example in this regard is the *knapsack problem*, which has been intensively investigated in the past [KPP04]. Suppose that we are given a set of n items, each having both a weight w_i and a value v_i. Furthermore, we are given a virtual knapsack with a finite weight capacity c. The problem now is to find a collection of items, in other words a non-negative integer number x_i for each item, such that they all together fit into the knapsack and at the same time maximize the total value. Our goal is therefore to solve the following optimization problem:

$$\max \sum_{i=1}^{n} v_i x_i$$
$$s.t. \sum_{i=1}^{n} w_i x_i \leq c$$

While this problem may appear to be limited to a very narrow field of applications, it turns out that there are many areas where problems have to be solved that are variants of the knapsack problem. Some of these problems are the optimization of cutting raw material to minimize the waste, the selection of investments and portfolios, the selection of asset-backed securitization, and generating keys for the Merkle–Hellman and other knapsack cryptosystems [KPP04].

In the literature, one can find many more famous examples of discrete optimization problems, some of which are connected to a very particular domain. Due to their inherent discrete structure, two domains that almost naturally reveal a multitude of discrete optimization problems are graphs and grids.

1.1.1 Graphs

A *graph* is an abstract structure that is usually represented by a set of nodes V and a set of edges E [Wes00]. The edges describe relations between pairs of nodes and can either be undirected or directed. An undirected edge can be considered as a symmetric relation between two nodes and is usually visualized as a straight line between the nodes. A directed edge, that is in the literature also referred to as arc, is an unsymmetric relation and is usually visualized as an arrow, thereby illustrating that a transition between the nodes is only possible in one direction. Graphs play an important role in navigation and allow an easy mapping of a real street network to a computational model. Usually, the nodes V and the edges E represent the intersections and road segments respectively. Many algorithms on graphs do not only need the structure of the graph given by sets of nodes and edges, but also a real number per edge that expresses the cost or length of the edge and is often referred to as *weight*. For navigational purposes, the weight that is assigned to each edge is chosen according to the distance, the traveling time or any similar metric that expresses the cost to get from one node to an adjacent node. An undirected graph would be used to model a road network in which all road segments are accessible in both directions, a directed graph would be used if there are any road segments involved that can be accessed in only one direction exclusively. Within a directed graph, it is however no problem to model roads that are accessible in both directions, which is simply achieved by connecting adjacent vertices with pairs of edges with opposite direction. To allow a better understanding, we want to give a more formal overview on the various graph types in the following.

Types of Graphs

As we have already indicated above, depending on the types of edges that are used to connect pairs of vertices, a graph can either be directed or undirected. In the literature, the word 'undirected' is, however, usually omitted when referring to an undirected graph. We want to further distinguish between graphs with and without multiple edges, since for our proposed methods only those graphs without multiple edges are of special interest. A formal differentiation between the types can be achieved by a proper analysis of the set of edges E. This gives a detailed insight into the structure, because a graph is

- undirected without multiple edges if $E \subseteq \{\{v_i, v_j\}\}; v_i, v_j \in V$ denoting that E is a subset of the 2-element subsets of V.
- directed without multiple edges if $E \subseteq \{(v_i, v_j)\}; (v_i, v_j) \in V \times V$ denoting that E is a subset of the Cartesian product $V \times V$.
- undirected with (combined) multiple edges if E is a *multiset* over $W = \{\{v_i, v_j\}\}$; $v_i, v_j \in V$ and therefore a function $E : W \to \mathbb{N}_0$.
- directed with (combined) multiple edges a multiset over the Cartesian product $V \times V$ and therefore a function $E : V \times V \to \mathbb{N}_0$.

In Figure 1.1 we depict some toy examples for each mentioned graph category. For the proposed methods in this thesis, graphs without multiple edges are of special interest since their structure naturally resembles the structure of the problems that we want to solve.

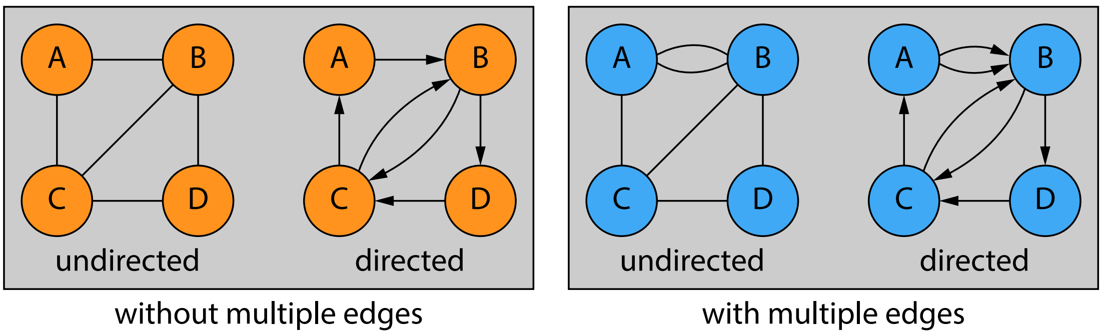

Figure 1.1: Examples of graphs. The vertices of those types that are of special interest to our proposed approaches are highlighted in orange color.

Problems on Graphs

A classical problem in graph theory is the computation of a path between two nodes. According to the definition [Wes00], a path is a sequence of edges that connect a sequence of vertices such that all vertices are distinct from one another. Interestingly, while the shortest path between two nodes can be computed efficiently (e.g., with Dijkstra's shortest path algorithm), finding the longest path between two nodes is a challenging task for which no approach with polynomial runtime is known (cf. Figure 1.2). Both problems are, however, indeed discrete since the problem formulation precisely asks to find a subset of edges that together fulfill a particular requirement. An edge is therefore completely inside or completely outside of a particular solution set, but cannot belong to the solution only fractionally. One could therefore formalize the problem by introducing a binary variable for each edge and define the solution to be represented by those variables that get a value of 1 assigned, while all other variables would get a value of 0.

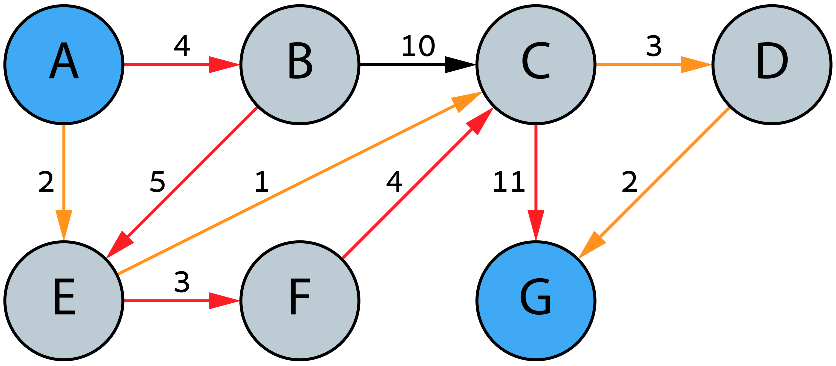

Figure 1.2: Shortest path (orange) of length 8 and longest path (red) of length 27 from node A to node G.

Another standard problem on graphs is that of finding a *minimum spanning tree* (MST). An MST consists of a subset of all edges of a graph that connect all vertices such that the total edge weight is minimum [Wes00]. In Figure 1.3 we show an example graph together with its minimum spanning tree.

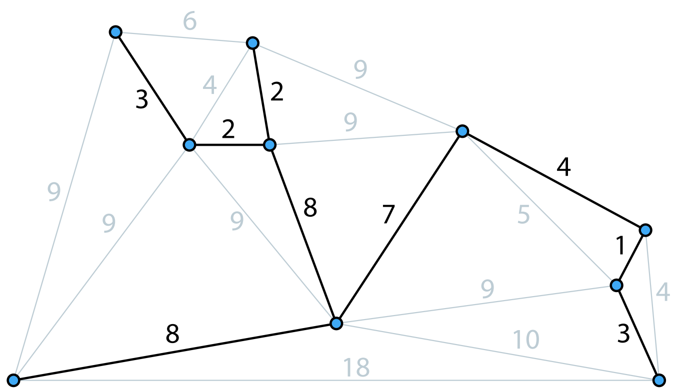

Figure 1.3: A graph together with its minimum spanning tree (MST). All edges belonging to the MST are drawn using bold lines.

Graphs in This Thesis

In this thesis, three peer-reviewed papers are presented in which the authors propose novel methods to advance the computer graphics research area. The proposed methods produce substantially distinct imagery output, but have one central aspect in common: They all use a graph to model the core problem and aim at finding a solution by formalizing the problem as a discrete optimization task.

In detail, the first publication with the title *Automatic Generation of Tourist Brochures* utilizes a graph in a similar manner to navigation and therefore directly maps a given street layout to a directed graph with edge weights that represent the travel costs. Within an iterative algorithm, the proposed method aims at computing a sub-graph that can be printed without visual clutter on a static map but at the same time allows an efficient travel between a given number of points of interest (POIs) without considerable detours.

The second publication, *Dynamic Path Exploration on Mobile Devices*, tries to improve on the idea of the first paper by enhancing the functionality to a dynamic output. Similar to the first publication, a graph that resembles the structure of a street layout is used as the central computational domain. However, the assigned weights do not represent the travel costs since the central idea of the method is the computation of a high-quality path, where quality is measured as the proximity to highly ranked tourist attractions.

Finally, the last publication *String Art: Towards Computational Fabrication of String Images* also chooses a graph as the core domain and also provides a continuous sequence of edges (i.e. a path) as output, but with a completely different goal in mind. Instead of being relevant for tourists like the computed paths from the first two publications, the edges of the provided path are interpreted as moving instructions for an industrial robot that follows the path to span a continuous piece of thread between pins that are distributed on a circular frame. All the spanned edges then fuse to a perceptible image that reassembles an input image as best as possible. While the first two publications produce imagery for navigational purposes, the output of the last mainly fulfills an artistic function.

1.1.2 Grids

A *grid* is a canonical structure that tessellates the Euclidean space into a set of congruent parallelotopes [TSW99]. Although we focus on 2d rectilinear and Cartesian grids in this thesis, we first want to give an overview over several types of 2d grids.

Types of Grids

Grids can be categorized into an *unstructured* and a *structured* type [TSW99]. Unstructured grids correspond to a tessellation of the Euclidean space using simple shapes like triangles, thereby producing an irregular pattern. Since the particular elements cannot be indexed in a straightforward manner, one needs some data structure that encodes the connectivity of the elements as well as the positions of the vertices. In contrast, structured (in the literature also referred to as regular) grids allow an indexing of the cells using 2d integer coordinates (i, j).

Structured grids can be further categorized into the following types:

- Cartesian grids where the cells are unit squares
- Rectilinear grids, which tessellate the Euclidean space using rectangles and parallelepipeds that are not necessarily congruent to each other.
- Curvilinear grids, whose combinatorial structure is very similar to a regular grid. However, the cells are not rectangles or squares but quadrilaterals.

Grids play an important role in layout determination by supporting designers in the following way:

1. For a closed area, they limit the available positions for element placement to a finite number and in turn reduce the complexity of layout design.
2. They simplify the identification of a coherent layout in which elements are placed in columns and rows of connected grid cells.

In Figure 1.4 we depict some grid type examples. For the proposed methods in this thesis, Cartesian and rectilinear grids, for which the tessellation with parallelotopes is achieved using rectangles and unit squares, respectively, are of special interest since their structure directly follows some approved guidelines for layouting and therefore easily allows us to model optimization problems to find a coherent layout for the rectangular entities we want to position on the canvas.

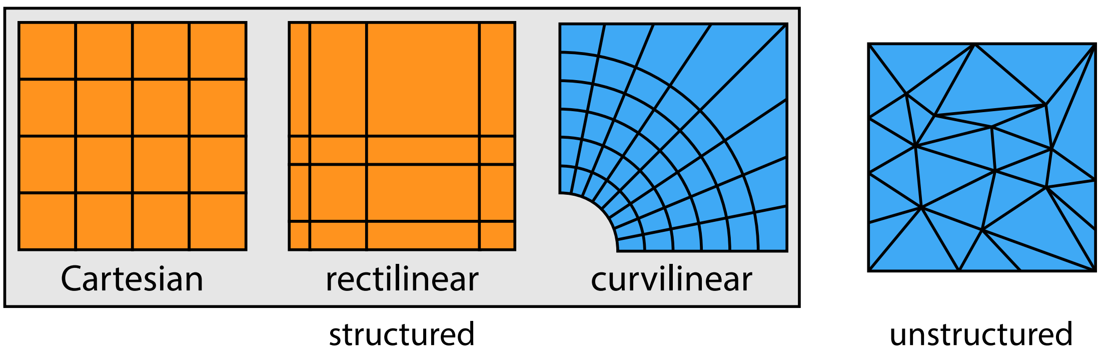

Figure 1.4: Examples of 2d grid types. Those types that are of special interest to our proposed approaches are highlighted in orange color.

Problems on Grids

There is a wealth of problems that can be formalized using a grid as the central computational domain. Some of these problems assume an infinite tessellation of the Euclidean space while others target a bounded region like a chessboard.

One famous combinatorial problem in this regard is the eight queens puzzle, which describes the problem of placing eight chess queens on an 8×8 chessboard so that no two queens threaten each other (cf. Figure 1.5). A feasible solution therefore is not allowed to have two queens on the same row, the same column or on the same diagonal. The problem is not limited to chessboards and can be extended to the problem of placing n queens on an $n \times n$ grid. It has been shown that this n-queens problem has a solution for all natural numbers n with the exception of $n = 2$ and $n = 3$ [BS09].

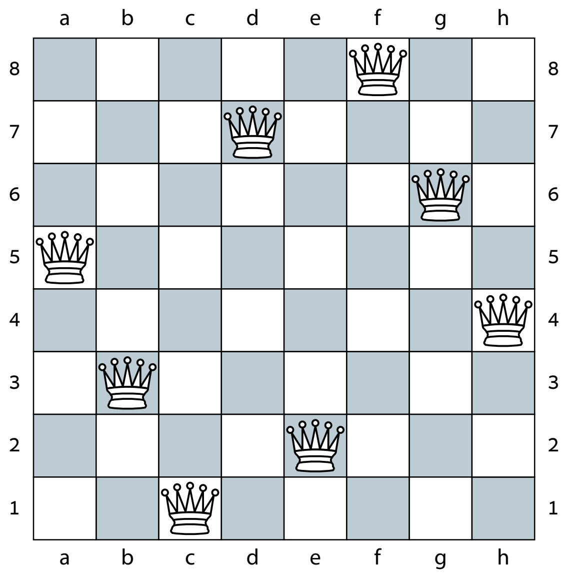

Figure 1.5: A valid solution to the eight queens problem. No two queens share a row, a column or a diagonal.

Grids in This Thesis

The first two publications presented in this thesis utilize grids to find a coherent layout of 2d information blocks. In detail, the method proposed in the first publication, *Automatic Generation of Tourist Brochures*, computes a layout of static and therefore printable maps. These maps consist of an overview map and several detail maps, which show detailed information about one or more POIs and which are placed around the overview map. The placement of these detail maps is done by following a set of approved design guidelines. By the utilization of grids, taking care of these guidelines is simplified since the problem of finding a coherent layout can be formalized as an integer program that naturally produces less cluttered layouts, in which the elements are positioned in columns and rows of the grid.

The second publication, *Dynamic Path Exploration on Mobile Devices*, follows similar design guidelines but provides a dynamic visualization as output. This output is particularly designed to be shown on hand-held devices like smart phones or tablets and can assist tourists to explore an unfamiliar environment. The placement of the information blocks in the visualizations is thereby not only performed by utilizing the available space provided by the grid on the map but due to the dynamic nature it also takes advantage of time as another powerful dimension to expand the solution space. This allows an overlap-free placement of several rectangular information blocks in one particular grid cell but in different periods in time.

1.2 Problem Statement

The problems that are tackled in the presented publications can be roughly categorized into two major research areas: navigation and arts. Although these two topics seem to have only a small overlap, the computational models and algorithms that are used to approach a variety of problems of either topic turn out to be very similar (cf. Section 1.1.1). Despite this overlap in the computational domain, in this section the problems from each research field are presented separately not from a computational but from a high-level point of view.

1.2.1 Research Problems in Navigation

Navigation is a very actively investigated research area and there is a lot of work published in a diversity of communities like geographic information systems (GIS), algorithms and data structures, and computer graphics. The increasing requirement for faster algorithms and better visualizations is motivated by the increasing number of mobile devices that are used as navigation systems as well as the establishment of new high-precision satellite-based positioning systems like the GALILEO system, which is currently being installed and which provides an accuracy up to a view centimeters for paying customers. With this increasing precision and demand, the field of navigation is confronted with new fields of application and a lot of challenges that are the driving factor for research and novel developments:

- **Presentation problem.** People usually navigate when they are outside and use their smartphone to retrieve navigational instructions. Due to the limited space on the screen, researchers and developers are confronted with the problem of choosing the most important from a wealth of information, and to choose the right visualization techniques to guide the user.

- **Exploration.** Especially tourists often do not want to navigate on the shortest path to one dedicated location, but they rather want to explore the environment in an unfamiliar region. However, this might fail without any meaningful instructions.
- **Static and dynamic output.** A map that is optimized for a large DIN A3 printout and which performs reasonably well in practice might completely fail when it is visualized on a smartphone. It is therefore of great interest to understand the characteristics of the output devices and the circumstances under which the navigational instructions will be used.
- **A to N and N to A.** Navigational problems can be much more complex than a simple route computation from point A to point B. For example, one might want to provide a coherent printable map to give routing instructions how to reach a certain number of locations starting at one particular location (A to N), or the problem might be to print a map to explain how the point A can be reached from various other locations (N to A).

1.2.2 Problem Statement of This Thesis concerning Navigation

In the following we present the fundamental problems of the first two publications of this thesis. We establish new research questions for both a static and a dynamic setup and will later propose our strategies to model and solve the problems efficiently. Both publications consist of two parts, a route identification part, in which we identify the relevant street segments that act as the atomic elements for all routing instructions, and a layout part, in which we aim to find an optimal placement of information about the POIs, which act as the corner points for all navigation tasks.

Automatic Generation of Tourist Brochures In our first presented publication, the main research problem concerning navigation is to automatically generate a static printable map containing efficient navigational instructions for traveling between a set of preselected POIs. Many approaches have been proposed how to tackle a variety of such routing problems. Some of these methods focus on efficient visual representations of routes from A to B. Agrawala et al. [AS01] created route maps that are similar to hand-drawn sketches. Karnick et al. [KCJ+10] introduced a system to visualize routes using local detail lenses automatically placed on the canvas. Other methods focus on the visualization of more complex route networks to give navigational instructions from many points to one dedicated destination (N to A). Kopf et al. [KAB+10] proposed a system for selecting and laying out the important roads based on mental representations of road networks. In contrast, our novel research question asks to find a coherent visualization that provides efficient instructions for traveling between many points from where each can act both as start point and as end point (N to N).

A naive approach to tackle this problem would be to take the union of all shortest paths between each pair of POIs and to print all contained route segments at a reasonable scale on a static map. However, while this strategy would work pretty well for a reasonably small number of destinations, an increase of this number would successively increase the amount of visual clutter and would therefore make the readability more and more difficult and the applicability of this map for navigation negligibly low. The main problem is therefore how this cluttered graph can be simplified to allow both a better readability and an efficient travel without considerable detours.

The information that a user of a printed map needs to decide whether he or she should head for a particular destination gives good reason to augment the map with essential information (e.g., category, rating, opening times) about the POIs. This requirement establishes the second central research problem and asks for a layout in which the information is presented. Finding layouts for a diversity of elements is a well-studied problem, and several design principles were proposed in recent work that can serve as guidelines for algorithmic solutions [ALB11]. For instance, one principle relies on the observation that a layout is better to read if coherent pieces of information are placed in proximity to one another. Another principle that we want to consider states that the map pins corresponding to the POIs should resemble as best as possible the arrangement of the respective pieces of information they represent.

Dynamic Path Exploration on Mobile Devices While printed maps are well-proven guides used for navigation and are still used very often in the field, the low price of smartphones and tablets together with the ubiquitous availability of cheap or even free data connections like open wireless networks has turned them into a true alternative used for navigation and to retrieve up-to-date information. Nowadays, people use their mobile devices together with some tourist application to retrieve information about the points of interest in their environment. These apps usually allow limiting the search result to a certain number of categories (e.g., museums, bars) and present the suitable POIs in ascending order with respect to the distance to the current or some predefined position. In turn, the mobile device is used as a navigation system to guide the user on the shortest path to the chosen destination.

We could identify two major drawbacks in this procedure that lead to new challenging problems and which were the main motivation for our work:

1. While the shortest path guarantees the user to reach the destination as fast as possible, it can not guarantee to be interesting for the user as well. In fact, there might be a path that is only slightly longer than the shortest path that the user would prefer since it would allow her to pass by some other POIs, making also the travel to the destination an interesting experience.

2. Common applications allow the user to choose the destination and to navigate only in a strictly separated manner. However, users meeting a region for the first time may not want to directly travel to some dedicated POI but they are more interested in exploration.

The problem deduced from drawback (1) is the identification of an interesting walking path through a set of POIs that should not exceed some maximum length constraint. If the user wants to define some dedicated POI as destination, the problem is similar but is augmented with the constraint that the computed path has to end at this POI. Note that this problem cannot be easily solved by well-known methods like Dijkstra's shortest path algorithm but it is more related to the NP-hard longest path problem. Previously published methods tackle the high-quality path-computation problem using dynamic programming [LWY$^+$10], adapting a shortest path algorithm [KW14], and genetic algorithms [MW06]. In contrast, we formalize the problem as a binary integer program (BIP) and can show that our approach significantly outperforms the previous work both according to runtime as well as to the quality of the result.

Given an interesting path and a set of POIs as a solution to the first problem, drawback (2) leads to the question how the path can be visualized in a way such that the focus lies not only on the path itself but also on its environment, allowing the user to better explore an unfamiliar region. In our setup we use an orthographic view onto a Mercator-projected map since people are already skilled to interact with similar maps from using Google Maps or Bing Maps. The problem is therefore how we can improve the map cutouts that represent the moving viewport onto the map within a dynamic visualization that is finally presented to the user.

Finally and following similar guidelines as in the preceding paper, the last problem is to find a suitable layout in which important information about the POIs (e.g., ratings, opening hours) is shown. The novelty of this problem referring to our setup lies in the dynamic nature of the generated visualizations, which allow a positioning of elements not only in the 2d domain of the map, but ask for a placement in a 3d space that is spanned together with the time axis as another powerful dimension and which enables us to show certain elements at the same position but in different instances of time.

1.2.3 Research Problems in Arts

Creating artistical-looking images has a long tradition in the computer graphics community, and many approaches for the abstraction and creation both for digital as well as for physical purposes have been proposed. Despite the diversity, these techniques usually have one aspect in common: there is some input example (often just an image), and there is a computational pipeline that transforms the input into a specific artistic style, appearance, or a physical phenomenon.

The development of novel approaches to create artistical-looking output is still a very active research field. In recent years, researchers came up with very creative approaches that can be assigned to a (non-exhaustive) number of categories:

- **Abstractions and conversions**. Methods that fit into this category take an image as input and aim at a transformation that lets the output mimic some artistic technique, e.g., to let it look like it was drawn by some particular artist.

- **Using shadows and perspective**. Other approaches aim at the creation of images by controlled shadow casting. The core problem in this category is to find an arrangement of existing objects or to compute some fabricable 3d objects that either cast a desired shadow or depict some particular shape when observed from a specific position.

- **Using light and optics**. Yet another set of methods aims at the generation of images from reflections or other light effects, especially by fabricating surfaces of objects with controlled appearance and reflectance properties.

- **Reliefs and fabrication**. Reliefs are yet another technique for the creation of abstracted physical images.

1.2.4 Problem Statement of This Thesis Concerning Art

Although the method for automatic string art creation that we present does not claim to generate art, it is indeed an automation of a known technique that generates artistical-looking results from existing input samples. Insofar, the problem is not novel but was already tackled by a number of artists [Vre16, Var17]. Although there exist algorithmic solutions, the physical results are usually crafted by hand.

To the best of our knowledge, we present the first formal treatment of the problem and an automatic computational fabrication pipeline to generate string images that resemble a gray scale image given as input. We describe the fundamental problems to automatically generate string images in the following.

String Art: Towards Computational Fabricaton of String Images Given a gray-scale image as input, the main problem of string art is how this image can be reassembled using a continuous piece of string. The challenge thereby is the proper formulation of an optimization problem that computes the path of the string.

The research question that we establish therefore asks how to compute the string path around the given pins and to fabricate the string images automatically using a robot. While the pins can be placed in any pattern, we focus on a setup where pins are distributed along a circular frame that encloses a canvas. In addition, we would like the image to be generated by a single string without cutting it.

There are many parameters and design choices leading to very different optimization problems. For example, the selection of parameters, like the number of pins and the image resolution used in the computation, drive the scale of the problem. The image formation model influences the complexity of optimization.

In later sections we will propose our strategies to model and solve the raised problems efficiently.

1.3 Research Goals

Based on the identified problems, we can formulate a number of goals, both for the research fields as well as for our proposed approaches.

1.3.1 General Research Goals in Navigation

The problems that we outlined in Section 1.2.1 already give a good indication which goals should be pursued by researchers working in the fields of navigation. While traditional maps are an approved tool that can provide reliable navigational instructions for a variety of routing problems, they usually contain a lot of information that is not needed for the particular use cases which makes an identification of the important content unnecessarily difficult. A central goal for the research area of navigation should therefore be a careful choice of suitable visualization methods as well as a reasonable identification of the target user, the target medium (static or dynamic) and the circumstances under which the system under development might be used.

1.3.2 General Research Goals in Arts

Due to the large spectrum of artistic approaches and directions, it is hard to define general goals for this research area concerning particular problems that should be tackled. With the falling prices and raising popularity of 3d printers and other custom-oriented fabrication devices, one can, however, point out a general goal that is crucial to keep in mind to follow this trend: fabrication-aware considerations. To be able to make a fabrication of the final piece of art possible, it is of high importance to have the characteristics of the fabrication process early in mind and to include all necessary requirements into the computational model. For instance, we included the diameter and the opacity of the used thread into our computational model to make a fabrication of our string art results possible.

1.3.3 Research Goals of Our Proposed Approaches

Based on the problems that we outlined in Section 1.2, we can point out a number of goals that we will cover in this thesis. In Section 1.4 we will take a closer look on the methodological approach that was taken to reach those goals.

Automatic Generation of Tourist Brochures

The fundamental goal of our first publication is to provide a framework that allows the automatic generation of static printable maps, also referred to as tourist brochures. These brochures provide easily readable navigational instructions between a set of preselected POIs. This selection is done by the user and allows the generation of a fully customized tourist brochure that only shows potential destinations and attractions the user is really interested in. Since one of our main goals is a good readability, we conducted a user study in which we could prove the applicability of our generated output. As we have already outlined in the problem statement in Section 1.2.2, our approach consists of two parts, a route identification part and a layout part, and we pursue the following goals for these parts to come up with a comprehensible output:

For the route identification, our goal is to develop an algorithm for the computation of a subset of the graph corresponding to the street network the POIs are embedded in. This subgraph has to meet the following requirements:

1. It must allow traveling between any pair of POIs without any considerable detour. What "considerable" means in this context can be chosen by the user. E.g., when the user chooses a value of 20%, the result graph must contain a path between any pair of POIs that is not more than 20% longer than the shortest path between those POIs in the original graph.

2. It must be printable on a static map without visual clutter, which means that it must be obvious from the static representation how one can navigate from one POI to the other.

In order to comply with approved design guidelines, we further pursue the following goals for the second part, in which we ask for a layout of rectangular entities that we refer to as *detail lenses* and which encapsulate the essential information about the POIs:

1. The order of the detail lenses should resemble the order of the corresponding POIs on the map.

2. POIs that are located close to each other should be grouped together and get a multi-POI detail lens.

3. Detail lenses should be positioned on the tourist brochure as close as possible to the corresponding POI, or in the case of a multi-POI detail lens, as close as possible to the corresponding group of POIs.

Dynamic Path Exploration on Mobile Devices

The goals that we pursue for our second publication are in several aspects similar to the goals we mentioned above for the creation of tourist brochures. Again, we aim at a system with which the user can automatically generate visual output that provides highly customized navigational instructions. In contrast, this time we target, as we have already outlined it in the problem statement in Section 1.2.2, not a static printable but a dynamic output that is designed to be shown on mobile devices like smartphones or tablets. What is, however, still similar to our preceding publication is the division of the whole problem into two parts: a route identification part and a layout part, and we want to outline the particular goals for either parts in the following.

For the first part, our goal is to identify an interesting high-quality route for the user. Quality is a quite subjective property and an application-dependent metric, but there exist approved methods how it can be measured. In recent work, quality was determined by the analysis of photo locations in proximity to the path or the retrieval and weighting of ranking values according to the distance to surrounding POIs. We follow strategies that are similar to the latter mentioned approach and use an online database (Yelp) to retrieve the corresponding ranking values. The particular properties of the path that we aim for are then as follows:

1. It should pass by as many highly ranked POIs as possible, thereby being interesting for the user. The categories of the POIs can again be freely chosen.
2. It must not be longer than a maximum predefined distance.

For the second part, we follow the concept of rectangular entities called detail lenses, which encapsulate fundamental information about the POIs. Our goal is to compute a coherent layout of these detail lenses. The properties of this layout can be listed as follows:

1. To follow approved design guidelines that we already obeyed in the preceding publication, we want to position the detail lenses as close as possible to the corresponding map pins to make an identification of associated pieces of information as easy as possible.
2. The detail lenses should not overlap mutually or with any part of the map that is crucial for navigation.

String Art: Towards Computational Fabricaton of String Images

Since our third publication does not follow a navigational but an artistic intent, the goals that we pursue for our third publication differ significantly from the two preceding papers. In detail, we aim at a framework that takes a gray-scale image as input and outputs at the very end a fully fabricated string art image. To this end, we pursue the following steps towards a comprehensible output:

1. Meeting a set of requirements for digital fabrication (e.g., thread thickness, thread opacity, frame diameter, pin distribution), our main goal is to compute a connected set of strings that best reassembles the input image.
2. As an intermediate result, we aim at a prediction image that shows how the final fabricated result will look like.
3. Finally, we aim at a hardware setup including a high-precision industrial robot with which we can produce the string art images automatically.

1.4 Methodology

During our research work we seek a clear and structured workflow that can be roughly divided into the following steps:

1. **Problem identification and definition.** In the beginning of each project we seek to identify an interesting research problem. To guarantee the novelty and significance of an idea, this is always done in alignment to the state of the art.
2. **Literature research.** An important part of our work is a thorough literature research, for which we consult the digital libraries of the Association for Computing Machinery (ACM), the Institute of Electrical and Electronics Engineers (IEEE) and Eurographics (EG), to name just a few. This enables us to confirm the importance of an identified research problem and to put into the right context.
3. **Implementation.** Although we aim at a clean and preferably object-oriented implementation of our ideas, we do not claim to produce high-performance market-ready code. Our implementations can more be seen as a proof-of-concept.
4. **Evaluation.** We evaluate our work both in a quantitative and a qualitative manner. For the first category, we profile our approaches and compare them to the state-of-the-art methods concerning runtime, memory consumption and also usability. For the latter category, we conduct user studies. By the analysis of the quantitative and qualitative results, we can justify the significance of our approaches.
5. **Identifying new research directions.** Especially the results from the user studies and the discussions with end users often give us the possibility to refine the problem definition and to optimize our approaches. These discussions also let us identify new problems and give us the opportunity to strike a new research direction.

In the following sections, we want to highlight some of the methodological aspects of our presented publications.

1.4.1 Automatic Generation of Tourist Brochures

We formalize the problem of finding a clutter-free subgraph that still allows an efficient travel between each pair of POIs as a greedy iterative procedure. We start out with the graph consisting of all shortest paths. We then pursue two goals: We want to keep the driving/walking time as low as possible while at the same time we want to remove as many road sections as possible. Note that these requirements are contradictory, and the task is therefore a balancing task, since each removal of a road segment increases the distances between the POIs. In order to quantize which sections should be removed, we define an energy function that reflects the quality of the residual graph. Then, in each iteration, we remove the road section that provides the smallest energy increase. We terminate our algorithm when either the distances between the POIs get larger than a predefined value, or when the residual graph is a spanning tree, in which case a removal is impossible without splitting it into two connected components, making a travel between certain pairs of POIs impossible.

For the computation of a coherent layout, we define an energy term that considers a number of design principles. We want to place the detail lenses as close as possible to the corresponding map pins on the overview map, but at the same time we want the order of all detail lenses to resemble the map pins on the map. We formalize the whole problem as a binary integer program (BIP) and use a highly optimized integer programming solver (Gurobi [Gur13]) to find a good feasible solution. In Figure 1.6, an example tourist brochure that is generated using our system is shown.

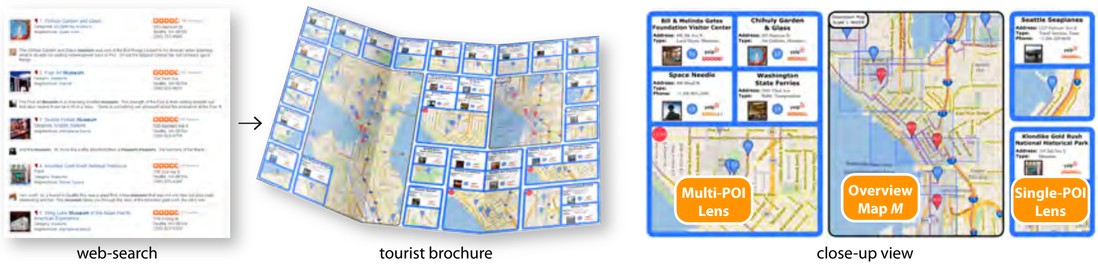

Figure 1.6: Given a set of points of interest (POIs) retrieved from an online database, our system automatically generates customized tourist brochures that provide both navigational instructions as well as information about the contained destinations.

1.4.2 Dynamic Path Exploration on Mobile Devices

The path that we want to compute should pass by as many highly ranked POIs as possible. To this end we assign weights to all street segments according to the sum of ranking values of all incident POIs. We seek for a path that maximizes the total weight while at the same time should stay below a maximum distance. Since this task cannot be tackled with approaches like Dijkstra's shortest path algorithm, we formalize it as a binary integer program (BIP) and solve it with Gurobi. We can show that, although the problem is NP-hard, we can find a near-optimal solution within a few seconds.

As a next step, we seek for an optimization of a path for the map cutouts that represent the viewport of an orthographic camera that is used in the dynamic visualizations which are presented to the user as final output of our approach. We assign 2d Gaussian functions with amplitudes that correspond to their ranking values to the POIs. The resulting mixture of Gaussian is then handled as an indicator function to highlight interesting regions. Since we want to emphasize these regions in our dynamic visualizations, but also want to keep the path smooth, we compute the path for the map cutouts as a B-spline running through the local maxima of this function while at the same time keeping the position of the user always within the screen.

Finally, we define an energy function to quantize the quality of a positioning of detail lenses. Again, this energy function considers a number of design principles we want to account for. We seek for a positioning of the detail lenses as close as possible to the corresponding map pins without any mutual overlap or any overlap of important information. Further, since we create a dynamic visualization, each detail lens should be visible as long as possible but should only appear in one continuous period of time. We formalize the positioning of detail lenses as a binary integer program (BIP) and again use Gurobi to find a good feasible solution. The final output is then shown on a mobile device like a smartphone or tablet. In Figure 1.7 the computational pipeline of our approach is shown.

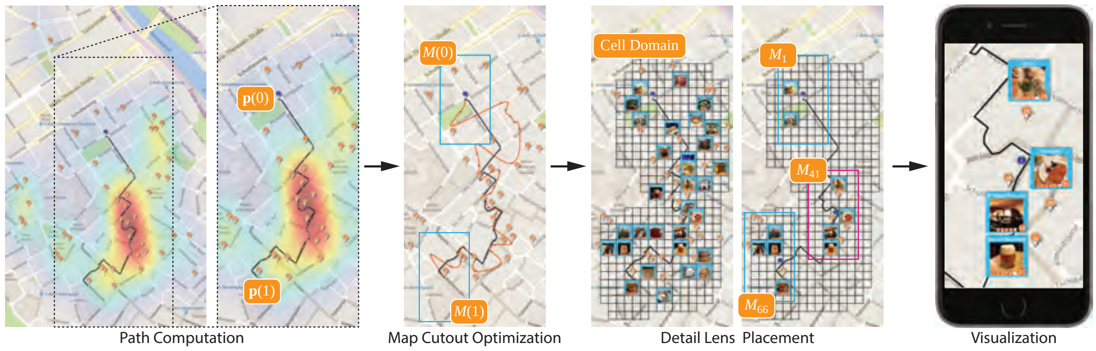

Figure 1.7: Given a set of points of interest (POIs) retrieved from an online database, our system automatically generates a dynamic visualization that enables the user to explore the environment.

1.4.3 String Art: Towards Computational Fabricaton of String Images

Due to the complexity of the problem of finding a continuous path that best reassembles the input image, we propose to divide the problem into two parts, an edge selection and a path extraction part. For the first part, we propose a greedy binary least squares solver that selects in each iteration the edge that decreases the residual most significantly. We continue until the residual cannot be decreased any more by the addition of an edge. Since a greedy approach like ours tends to deliver a suboptimal local minimum, we propose the following improvement. After our algorithm terminates, we run it again on the result, but do not add but remove that edge that can reduce the residual most significantly. This is then again continued until an improvement by removal is impossible. We propose to proceed in an alternating manner, so to add and remove edges until we are stuck in a minimum that can not be left, neither by the removal nor the addition of an edge.

In the next step, we extract a path from the set of selected edges. Since the graph that is defined by these edges is not necessarily Eulerian, which is however by definition a necessary condition to contain one continuous path, we have to add some auxiliary edges.

These edges do not influence the final fabricated result since they can be drawn on the outside of the canvas by moving the tooltip on an arc around the frame between the two corresponding pins.

For the fabrication, we utilize a highly accurate 6-axis industrial robot. In order to get proper tension on the string but at the same time avoiding a break of the thread, we use a tension regulator of a professional knitting machine. For convenience, we use aluminum bicycle rims for the frame and conventional wall hooks at the pin positions.

In Figure 1.8, the pipeline of our approach is shown.

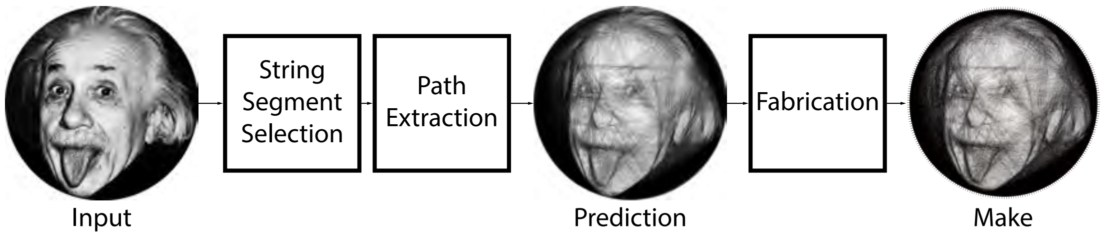

Figure 1.8: Given a gray scale image as input, we meet a number of computational requirements to identify a continuous path that best reassembles the input image. We use an industrial robot for fabrication.

1.5 Contributions

In the following we show how our publications could contribute to the scientific community. We give a short overview on the proposed approaches, point out the novelty of our research direction, and give a brief indication how we evaluated our findings to justify their significance.

1.5.1 Automatic Generation of Tourist Brochures

In our first publication we present a system for the automatic generation of tourist brochures with route information, which provide information about a set of points of interest in a particular region in a focus-and-context style by the use of detail lenses.

We contribute an algorithm for route-aware simplification of routing graphs that provide near-optimal routing directions on the one hand, and are simple and clutter-free on the other. While recent approaches focus on efficient visual representations of routes from A to B [AS01, KCJ+10], or provide instructions from many points to one dedicated destination [KAB+10], our novel research question asks to find a coherent visualization that provides efficient instructions for traveling between many points from which each can act both as start point and as end point (N to N).

We show that the routing graphs that are computed using our method match well with those selected by humans for traveling between arbitrarily distributed POIs. Moreover, we introduce a novel layout algorithm that optimizes a layout of a number of elements of different sizes in a discrete grid under specific neighborhood constraints. In the future we plan to extend the implementation of our system to real-time interactive route planning on desktop and mobile devices.

1.5.2 Dynamic Path Exploration on Mobile Devices

In the second publication, we present a system for dynamic route exploration on mobile devices. Our results give immediate guidance to a user by presenting both routing information for an attractive path whose environment contains many high-quality points of interest (POIs), as well as information about those POIs by the use of detail lenses. While previously published methods tackle the high-quality path computation problem using dynamic programming [LWY+10], adapting a shortest-path algorithm [KW14], and genetic algorithms [MW06], we formalize the problem as a binary integer program (BIP) and can show that our approach significantly outperforms the previous work both according to runtime as well as to the quality of the result.

In addition, we contribute an algorithm for optimizing the map cutouts which correspond to the viewports in the final visualization that is presented to the user. Moreover, we introduce a novel layout algorithm that computes a dynamic layout of rectangular entities in a discrete grid under specific constraints and present the results of a study that shows the significant acceptance rate of our method.

For future work, we want to extend the functionality of our framework. We think of a multi-layer approach in order to allow a visualization on different zoom levels and smooth transitions between the layers.

1.5.3 String Art: Towards Computational Fabricaton of String Images

In our last publication, we propose a novel method for the automatic computation and digital fabrication of artistic string images. This problem is not novel but was already tackled by a number of artists [Vre16, Var17]. However, to the best of our knowledge, we present the first formal treatment of the problem and an automatic computational fabrication pipeline to generate string images. We analyze the problem and derive a problem formulation as a non-linear binary least squares problem. We propose a hardware setup for the automatic digital fabrication of these images using an industrial robot that spans strings between pins. We also propose a greedy algorithm to compute an approximate solution to the optimization problem and demonstrate that the quality of our solution significantly outperforms other approaches. Finally, we demonstrate the applicability of our methods by generating and fabricating a set of real string art results.

In future work, we would like to extend our setup to experiment with more general pin placement, to generate 2.5d string art. Further, we would like to experiment with strings of different colors and more transparent strings. We are also interested to investigate the fabrication of customized shading systems using a similar process.

List of Figures

Bibliography

[Ake06] James R. Akerman. *Cartographies of travel and navigation.* University of Chicago Press, 2006.

[ALB11] Maneesh Agrawala, Wilmot Li, and Floraine Berthouzoz. Design principles for visual communication. *Communications of the ACM*, 54(4):60, apr 2011.

[AM10] Marc Alexa and Wojciech Matusik. Reliefs as images. *ACM Transactions on Graphics*, 29(4):1, jul 2010.

[AM11] Marc Alexa and Wojciech Matusik. Images from self-occlusion. In *Proceedings of the International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging - CAe '11*, page 17, New York, New York, USA, 2011. ACM Press.

[AS01] Maneesh Agrawala and Chris Stolte. Rendering effective route maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, pages 241–249, New York, New York, USA, aug 2001. ACM Press.

[BBAM12] Amit Bermano, Ilya Baran, Marc Alexa, and Wojciech Matusk. ShadowPix: Multiple Images from Self Shadowing. *Computer Graphics Forum*, 31(2pt3):593–602, 2012.

[BDY06] Ken Been, Eli Daiches, and Chee Yap. Dynamic map labeling. *IEEE transactions on visualization and computer graphics*, 12(5):773–80, jan 2006.

[BFSS07] Holger Bast, Stefan Funke, Peter Sanders, and Dominik Schultes. Fast routing in road networks with transit nodes. *Science (New York, N.Y.)*, 316(5824):566, apr 2007.

[BKB$^+$12] Ilya Baran, Philipp Keller, Derek Bradley, Stelian Coros, Wojciech Jarosz, Derek Nowrouzezahrai, and Markus Gross. Manufacturing Layered Attenuators for Multiple Prescribed Shadow Images. *Computer Graphics Forum*, 31(2pt3):603–610, 2012.

[BKSW07] Michael A. Bekos, Michael Kaufmann, Antonios Symvonis, and Alexander Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. *Computational Geometry*, 36(3):215–236, apr 2007.

[BMWW14] Michael Birsak, Przemyslaw Musialski, Peter Wonka, and Michael Wimmer. Automatic generation of tourist brochures. *Computer Graphics Forum*, 33(2):449–458, 2014.

[Bre65] J E Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

[BS09] Jordan Bell and Brett Stevens. A survey of known results and research areas for n-queens. *Discrete Mathematics*, 309(1):1–31, 2009.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[CBGS09] Wei-Chao Chen, Agathe Battestini, Natasha Gelfand, and Vidya Setlur. Visual summaries of popular landmarks from community photo collections. In *Proceedings of the seventeen ACM international conference on Multimedia - MM '09*, page 789, New York, New York, USA, oct 2009. ACM Press.

[CBHK09] David J Crandall, Lars Backstrom, Daniel Huttenlocher, and Jon Kleinberg. Mapping the world's photos. In *Proceedings of the 18th international conference on World wide web - WWW '09*, page 761, New York, New York, USA, apr 2009. ACM Press.

[CCLM13] Hung-Kuo Chu, Chia-Sheng Chang, Ruen-Rone Lee, and Niloy J Mitra. Halftone QR codes. *ACM Transactions on Graphics*, 32(6):1–8, nov 2013.

[Chi06] Luca Chittaro. Visualizing information on mobile devices. *Computer*, 39(3):40–45, 2006.

[CMS95] Jon Christensen, Joe Marks, and Stuart Shieber. An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics*, 14(3):203–232, jul 1995.

[Cor09] Thomas H Cormen. *Introduction to algorithms*. MIT press, 2009.

[CZ07] Xiao-Wen Chang and Tianyang Zhou. MILES: MATLAB package for solving Mixed Integer LEast Squares problems. *GPS Solutions*, 11(4):289–294, nov 2007.

[DFV+08] Angelo Di Iorio, Luca Furini, Fabio Vitali, John Lumley, and Tony Wiley. Higher-level layout through topological abstraction. In *Proceeding of the eighth ACM symposium on Document engineering - DocEng '08*, page 90, New York, New York, USA, sep 2008. ACM Press.

[ECMS96] Shawn Edmondson, Jon Christensen, Joe Marks, and Stuart Shieber. A General Cartographic Labelling Algorithm. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 33(4):13–24, dec 1996.

[Elb10] Gershon Elber. Ortho-Pictures: 3D Objects from Independent 2D Data Sets. In *Advances in Architectural Geometry 2010*, pages 175–192. Springer Vienna, Vienna, 2010.

[FM98] D. Fodness and B. Murray. A Typology of Tourist Information Search Strategies. *Journal of Travel Research*, 37(2):108–119, nov 1998.

[FP99] Jean-Daniel Fekete and Catherine Plaisant. Excentric labeling. In *Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit - CHI '99*, pages 512–519, New York, USA, 1999. ACM Press.

[GASP08] Floraine Grabler, Maneesh Agrawala, Robert W. Sumner, and Mark Pauly. Automatic generation of tourist maps. *ACM Transactions on Graphics*, 27(3):1, aug 2008.

[GHS06] Timo Götzelmann, Knut Hartmann, and Thomas Strothotte. Agent-Based Annotation of Interactive 3D Visualizations. In *6th International Symposium on Smart Graphics*, pages 24–35. Springer Verlag, 2006.

[GKAK16] B Galea, E Kia, N Aird, and P G Kry. Stippling with aerial robots. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, pages 125–134. Eurographics Association, 2016.

[Gur13] Gurobi Optimization, Inc. Gurobi Optimizer Reference Manual, 2013.

[Gur16] Inc. Gurobi Optimization. Gurobi Optimizer Reference Manual. http://www.gurobi.com, 2016.

[Hae90] Paul Haeberli. Paint by numbers: abstract image representations. *ACM SIGGRAPH Computer Graphics*, 24(4):207–214, sep 1990.

[Hir82] Stephen A Hirsch. An algorithm for automatic name placement around point data. *The American Cartographer*, 9(1):5–17, 1982.

[HJO$^+$01] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, pages 327–340, New York, New York, USA, 2001. ACM Press.

[HW73] Carl Hierholzer and Chr Wiener. Ueber die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren. *Mathematische Annalen*, 6(1):30–32, 1873.

[Imh75] Eduard Imhof. Positioning names on maps. *The American Cartographer*, 2(2):128–144, 1975.

[JLS+03] Charles Jacobs, Wilmot Li, Evan Schrier, David Bargeron, and David Salesin. Adaptive grid-based document layout. In *ACM SIGGRAPH 2003 Papers on - SIGGRAPH '03*, volume 22, page 838, New York, New York, USA, jul 2003. ACM Press.

[KAB+10] Johannes Kopf, Maneesh Agrawala, David Bargeron, David Salesin, and Michael Cohen. Automatic generation of destination maps. *ACM Transactions on Graphics*, 29(6):1, 2010.

[KCJ+10] Pushpak Karnick, David Cline, Stefan Jeschke, Anshuman Razdan, and Peter Wonka. Route visualization using detail lenses. *IEEE transactions on visualization and computer graphics*, 16(2):235–47, jan 2010.

[KIIF10] Takeshi Kurashima, Tomoharu Iwata, Go Irie, and Ko Fujimura. Travel route recommendation using geotags in photo sharing sites. *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 579–588, 2010.

[KIIF13] Takeshi Kurashima, Tomoharu Iwata, Go Irie, and Ko Fujimura. Travel route recommendation using geotagged photos. *Knowledge and Information Systems*, 37(1):37–60, 2013.

[KL11] Johannes Kopf and Dani Lischinski. Depixelizing pixel art. *ACM Transactions on Graphics*, 30(4):1, jul 2011.

[KPP04] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack problems.* Springer, 2004.

[KW14] Alexander Kachkaev and Jo Wood. Automated planning of leisure walks based on crowd-sourced photographic content. *Paper presented at the 46th Annual Universities' Transport Study Group Conference, 06-01-2014 - 08-01-2014, Newcastle, UK.*, (January):227–246, 2014.

[Laa18] Laarco. Laarco Studio | String Art, Algorithmic Art, Customized Portraits, 2018.

[LFC04] Man Luo, Ruomei Feng, and Liping A. Cai. Information Search Behavior and Tourist Characteristics. *Journal of Travel & Tourism Marketing*, 17(2-3):15–25, feb 2004.

[LFN04] Simon Lok, Steven Feiner, and Gary Ngai. Evaluation of visual balance for automated layout. In *Proceedings of the 9th international conference on Intelligent user interface - IUI '04*, page 101, New York, New York, USA, jan 2004. ACM Press.

[LGX+13] Anat Levin, Daniel Glasner, Ying Xiong, Frédo Durand, William Freeman, Wojciech Matusik, and Todd Zickler. Fabricating BRDFs at high spatial resolution using wave optics. *ACM Transactions on Graphics*, 32(4):1, jul 2013.

[LPD13] Thomas Lindemeier, Sören Pirk, and Oliver Deussen. Image stylization with a painting machine using semantic hints. *Computers & Graphics*, 37(5):293–301, aug 2013.

[LSS+14] Cheryl Lau, Yuliy Schwartzburg, Appu Shaji, Zahra Sadeghipoor, and Sabine Süsstrunk. Creating personalized jigsaw puzzles. In *Proceedings of the Workshop on Non-Photorealistic Animation and Rendering - NPAR '14*, pages 31–39, New York, New York, USA, 2014. ACM Press.

[LWY+10] Xin Lu, Changhu Wang, Jiang-Ming Yang, Yanwei Pang, and Lei Zhang. Photo2Trip: generating travel routes from geo-tagged photos for trip planning. *Mm*, pages 143–152, 2010.

[MP09] Niloy J Mitra and Mark Pauly. Shadow art. *ACM Transactions on Graphics*, 28(5):1, 2009.

[MSL+11] Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. Interactive furniture layout using interior design guidelines. *ACM Transactions on Graphics*, 30(4):1, jul 2011.

[MW06] Peter Mooney and Adam Winstanley. An evolutionary algorithm for multi-criteria path optimization problems. *International Journal of Geographical Information Science*, 20(4):401–423, 2006.

[OW07] A J Orman and H P Williams. *Optimisation, Econometric and Financial Analysis*, chapter A Survey o, pages 91–104. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[PDP+15] Daniele Panozzo, Olga Diamanti, Sylvain Paris, Marco Tarini, Evgeni Sorkine, and Olga Sorkine-Hornung. Texture Mapping Real-World Objects with Hydrographics. *Computer Graphics Forum*, 34(5):65–75, aug 2015.

[PF06] Bing Pan and Daniel R. Fesenmaier. Online Information Search. *Annals of Tourism Research*, 33(3):809–832, jul 2006.

[PJJ+11] Marios Papas, Wojciech Jarosz, Wenzel Jakob, Szymon Rusinkiewicz, Wojciech Matusik, and Tim Weyrich. Goal-based Caustics. *Computer Graphics Forum*, 30(2):503–511, apr 2011.

[PJJSH16] Romain Prévost, Alec Jacobson, Wojciech Jarosz, and Olga Sorkine-Hornung. Large-scale painting of photographs by interactive optimization. *Computers & Graphics*, 55(C):108–117, apr 2016.

[PS82] Christopher C Paige and Michael A Saunders. LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.

[QS08] Sanzheng Qiao and Sanzheng. Integer least squares. In *Proceedings of the 2008 C3S2E conference on - C3S2E '08*, page 23, New York, New York, USA, 2008. ACM Press.

[RMP11] Andreas Reichinger, Stefan Maierhofer, and Werner Purgathofer. High-quality tactile paintings. *Journal on Computing and Cultural Heritage*, 4(2):1–13, nov 2011.

[RR14] Maxim A Rylov and Andreas W Reimer. Improving label placement quality by considering basemap detail with a raster-based approach. *GeoInformatica*, 19(3):463–486, jul 2014.

[SABS94] Michael P Salisbury, Sean E Anderson, Ronen Barzel, and David H Salesin. Interactive pen-and-ink illustration. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques - SIGGRAPH '94*, pages 101–108, New York, New York, USA, 1994. ACM Press.

[SMPZ15] Roy Shilkrot, Pattie Maes, Joseph A Paradiso, and Amit Zoran. Augmented Airbrush for Computer Aided Painting (CAP). *ACM Transactions on Graphics*, 34(2):1–11, 2015.

[SMSW90] D. Snepenger, K. Meged, M. Snelling, and K. Worrall. Information Search Strategies by Destination-Naive Tourists. *Journal of Travel Research*, 29(1):13–16, jul 1990.

[SPG+16] Christian Schüller, Daniele Panozzo, Anselm Grundhöfer, Henning Zimmer, Evgeni Sorkine, and Olga Sorkine-Hornung. Computational thermoforming. *ACM Transactions on Graphics*, 35(4):1–9, jul 2016.

[SPSH14] Christian Schüller, Daniele Panozzo, and Olga Sorkine-Hornung. Appearance-mimicking surfaces. *ACM Transactions on Graphics*, 33(6):1–10, nov 2014.

[SS07] Peter Sanders and Dominik Schultes. Engineering fast route planning algorithms. pages 23–36, jun 2007.

[SSN+05] T Shiraishi, M Nagata N Shibata, M Nagata, N Shibata, Y Murata, K Yasumoto, and M Ito. A personal navigation system with a schedule planning facility based on multi-objective criteria. In *Proc. of 2nd Int'l. Conf. on Mobile Computing and Ubiquitous Networking*. Citeseer, 2005.

[STTP14] Yuliy Schwartzburg, Romain Testuz, Andrea Tagliasacchi, and Mark Pauly. High-contrast computational caustic design. *ACM Transactions on Graphics*, 33(4):1–11, jul 2014.

[SY17] Stela H Seo and James E Young. Picassnake. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17*, page 418, New York, New York, USA, 2017. ACM Press.

[TSW99] J F Thompson, B K Soni, and N P Weatherill. *Handbook of Grid Generation*. CRC-Press, 1999.

[Var17] Daniel Varga. String art. https://github.com/danielvarga/string-art, 2017.

[Vre16] Petros Vrellis. A new way to knit. http://artof01.com/vrellis/works/knit.html, 2016.

[WDB+07] Tim Weyrich, Jia Deng, Connelly Barnes, Szymon Rusinkiewicz, and Adam Finkelstein. Digital bas-relief from 3D scenes. *ACM Transactions on Graphics*, 26(3):32, jul 2007.

[Wes00] Douglas B West. *Introduction to Graph Theory*. Prentice Hall, 2 edition, sep 2000.

[WLSI14] Fangzhou Wang, Yang Li, Daisuke Sakamoto, and Takeo Igarashi. Hierarchical route maps for efficient navigation. In *Proceedings of the 19th international conference on Intelligent User Interfaces - IUI '14*, pages 169–178, New York, New York, USA, feb 2014. ACM Press.

[WPMR09] Tim Weyrich, Pieter Peers, Wojciech Matusik, and Szymon Rusinkiewicz. Fabricating microgeometry for custom surface reflectance. *ACM Transactions on Graphics*, 28(3):1, jul 2009.

[WXWX91] Xiaolin Wu, Xiaolin, Wu, and Xiaolin. An efficient antialiasing technique. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques - SIGGRAPH '91*, volume 25, pages 143–152, New York, New York, USA, 1991. ACM Press.

[YIC+12] Yonghao Yue, Kei Iwasaki, Bing-Yu Chen, Yoshinori Dobashi, and Tomoyuki Nishita. Pixel Art with Refracted Light by Rearrangeable Sticks. *Computer Graphics Forum*, 31(2pt3):575–582, 2012.

[YYW+12] Yi-Ting Yeh, Lingfeng Yang, Matthew Watson, Noah D Goodman, and Pat Hanrahan. Synthesizing open worlds with constraints using locally annealed reversible jump MCMC. *ACM Transactions on Graphics*, 31(4):1–11, jul 2012.

[Zal96] Antoine Zalatan. The determinants of planning time in vacation travel. *Tourism Management*, 17(2):123–131, mar 1996.

[ZLW+16] Haisen Zhao, Lin Lu, Yuan Wei, Dani Lischinski, Andrei Sharf, Daniel Cohen-Or, and Baoquan Chen. Printed Perforated Lampshades for Continuous Projective Images. *ACM Transactions on Graphics*, 35(5):1–11, jun 2016.

[Zor86] Steven Zoraster. Integer programming applied to the map label placement problem. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 23(3):16–27, 1986.

[Zor90] Steven Zoraster. The solution of large 0–1 integer programming problems encountered in automated cartography. *Operations Research*, 38(5):752–759, 1990.

[ZYZ+13] Yan-Tao Zheng, Shuicheng Yan, Zheng-Jun Zha, Yiqun Li, Xiangdong Zhou, Tat-Seng Chua, and Ramesh Jain. GPSView: A Scenic Driving Route Planner. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 9(1):1–18, 2013.

[ZYZZ15] Yizhong Zhang, Chunji Yin, Changxi Zheng, and Kun Zhou. Computational hydrographic printing. *ACM Transactions on Graphics*, 34(4):131:1——131:11, jul 2015.

CHAPTER 2

Papers

2.1 Automatic Generation of Tourist Brochures

Michael Birsak, Przemyslaw Musialski, Peter Wonka, and Michael Wimmer

Computer Graphics Forum (Proceedings of EUROGRAPHICS 2014), Volume 33, Issue 2, pages 449–458, 2014. DOI: 10.1111/cgf.12333

The definitive version of this paper is available at `http://diglib.eg.org/` and `http://onlinelibrary.wiley.com/`.

2.2 Dynamic Path Exploration on Mobile Devices

Michael Birsak, Przemyslaw Musialski, Peter Wonka, and Michael Wimmer

IEEE Transactions on Visualization and Computer Graphics, Volume 24, Issue 5, pages 1784-1798, 2018. DOI: 10.1109/TVCG.2017.2690294

The definitive version of this paper is available at `https://ieeexplore.ieee.org/`.

2.3 String Art: Towards Computational Fabricaton of String Images

Michael Birsak, Florian Rist, Peter Wonka, and Przemyslaw Musialski

Computer Graphics Forum (Proceedings of EUROGRAPHICS 2018), Volume 37, Issue 2, pages 263-274, 2018. DOI: 10.1111/cgf.13359

The definitive version of this paper is available at `http://diglib.eg.org/` and `http://onlinelibrary.wiley.com/`.

Michael Peter Birsak

Curriculum Vitae

Personal Data

Date of Birth	September 20, 1984
Place of Birth	Mistelbach an der Zaya
Nationality	Austria

Education

06/2012 **Master in Visual Computing**, *TU Wien*.
Focus on Rendering and Modeling (with distinction)

05/2009 **Bachelor in Media Informatics**, *TU Wien*.
Focus on Computer Graphics

06/2004 **General Qualification for University Entrance (Matura)**, *HTL Hollabrunn*.
Focus on Industrial Engineering with a Major in Computer Science (with distinction)

Experience

Research

2012-Present **Scientific Assistant**, Rendering and Modeling Group of the Institute of Visual Computing & Human-Centered Technology, TU Wien.
Research on Discrete Optimization for the Creation of Navigational and Artistic Imagery

Teaching at TU Wien

2012-2016 **Seminar aus Computergraphik**, (4 semesters).

2013-2016 **Seminar Wissenschaftliches Arbeiten**, (4 semesters).

2013-2016 **Internships (2), Bachelor Theses (2) and Master Theses (1)**.
Supervising assistant

Languages

German	**Native Language**
English	**Fluent**

Theses

Master Thesis	*Coloring Meshes of Archaeological Datasets*
Supervisor	Associate Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer

Haberlandtgasse 24/4/13 – 1220 Wien
+43 676 4250760 • michael.birsak@gmx.at

Description	This work was concerned with the automatic and artifact-free texturing of archaeological data.
Bachelor Thesis	*Juggling in a Virtual Environment*
Supervisor	Privatdoz. Mag.rer.nat. Dr.techn. Hannes Kaufmann
Description	This work was concerned with the effects and possibilities of juggling in a virtual environment.

Publications

Journals

Michael Birsak, Przemyslaw Musialski, Peter Wonka, and Michael Wimmer. *Dynamic Path Exploration on Mobile Devices*. IEEE Transactions on Visualization and Computer Graphics. May 2018

Michael Birsak, Florian Rist, Peter Wonka, and Przemyslaw Musialski. *String Art: Towards Computational Fabrication of String Images*. Computer Graphics Forum (Proc. EUROGRAPHICS 2018). April 2018

Przemyslaw Musialski, Christian Hafner, Florian Rist, Michael Birsak, Michael Wimmer, and Leif Kobbelt. *Non-Linear Shape Optimization Using Local Subspace Projections*. ACM Transactions on Graphics (ACM SIGGRAPH 2016). July 2016

Przemyslaw Musialski, Thomas Auzinger, Michael Birsak, Michael Wimmer, and Leif Kobbelt. *Reduced-Order Shape Optimization Using Offset Surfaces*. ACM Transactions on Graphics (ACM SIGGRAPH 2015). August 2015

Michael Birsak, Przemyslaw Musialski, Peter Wonka, and Michael Wimmer. *Automatic Generation of Tourist Brochures*. Computer Graphics Forum (Proc. EUROGRAPHICS 2014). April 2014

Others

Michael Birsak, Przemyslaw Musialski, Murat Arıkan, and Michael Wimmer. *Seamless Texturing of Archaeological Data*. Digital Heritage International Congress (DigitalHeritage). October 2013