FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Tangible Historical Maps

An approach towards customisable 3D printed historical maps

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Medieninformatik und Visual Computing

eingereicht von

Tanja Travnicek
Matrikelnummer 1126286

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Mitwirkung: Ing. Dr.techn. Peter Mindek

Wien, 9. Jänner 2017

Tanja Travnicek	Eduard Gröller

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Tangible Historical Maps

An approach towards customisable 3D printed historical maps

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Media Informatics and Visual Computing

by

Tanja Travnicek
Registration Number 1126286

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Assistance: Ing. Dr.techn. Peter Mindek

Vienna, 9th January, 2017

Tanja Travnicek   Eduard Gröller

Erklärung zur Verfassung der Arbeit

Tanja Travnicek
Hainburger Straße 47/18, 1030 Wien, Österreich

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, 9. Jänner 2017

Tanja Travnicek

Acknowledgements

I wish to thank various people for their contribution to this project; Mr Frank Reed, creator of “The Centennia Historical Atlas”, for the generous fair-use agreement on his data; Mr Tobias Perschon, for the access to his 3D printer and his valuable input throughout my work; Mrs Maria Perschon, Mr Michael Langowski and Ms Sabrina Burtscher, for proofreading.

Special thanks should be given to Ing. Dr. techn. Peter Mindek, my supervisor, for his professional guidance and constructive suggestions during the planning and development of this work.

Finally, I wish to thank my parents for their support and encouragement throughout my study.

Kurzfassung

Territoriale Veränderungen sind ein wichtiger Teil der Geschichte eines Landes, viele von ihnen enstehen durch Kriege, Aufstände oder durch andere politische Prozesse. Meistens werden solche Veränderungen mit Hilfe von großen Landkarten und verbalen Erklärungen unterrichtet. Da Landkarten auf ein einziges statisches Bild beschränkt sind, benötigen Schüler und Schülerinnen eine gute Vorstellungskraft um dem Unterricht zu folgen. Diese Arbeit stellt eine neue mögliche Unterrichtsmethode, basierend auf 3D gedruckten Modellen, vor. Das Ziel ist es, Veränderungen im Einflussbereich eines Landes in dynamischerer und interaktiverer Weise darzustellen.

Für diese Arbeit wurden zwei Prototypen angreifbarer historischer Landkarten produziert, welche aus standardisierten, zweidimensionalen Karten aufgebaut wurden. Territoriale Veränderungen sind in x- und y-Richtung wiedergegeben, während die z-Achse den zeitlichen Verlauf repräsentiert. Im Laufe der Arbeit wurde der Einfluss der derzeit noch beschränkten Möglichkeiten kostengünstiger 3D Drucker auf die vorgeschlagene Herangehensweise ersichtlich. Der Druckvorgang ist aktuell ein sehr zeitintensiver Prozess und Objekte sind aufgrund kleinerer Druckbereiche nur bis zu einer gewissen Größe realisierbar. Auf Grund dieser Einschränkungen wurde ein Programm entwickelt, mit Hilfe dessen, bestimmte Bereiche des 3D Modells verändert werden können. Als Ausgabe erhält man ein neues Modell mit einem benutzerdefinierten Detailgrad. Das Programm ermöglicht eine Vergrößerung oder Verkleinerung einzelner Abschnitte auf der z-Achse und schlägt außerdem auch Druckereinstellungen für ein ideales Ergebnis vor. Letzteres hilft Personen, welche sich bisher nicht ode rnur wenig mit 3D Druck befasst haben, bei der Erstellung von angreifbaren historischen Karten.

Zusammenfassend ist der vorgeschlagene Ansatz besonders dann sinnvoll, wenn nur eine kleine Anzahl an Ländern mit vielen, kurz aufeinander folgenden territorialen Veränderungen abgebildet werden soll.

Abstract

Territorial shifts are a vital part of a country's history, mostly caused by wars, uprisings or other political processes. The most commonly used method to teach that matter, is to bring huge pull down maps to class and give verbal explanations while pointing at the map. Those maps are limited to one static image and therefore need a significant amount of imagination on the student's side. This thesis suggests a new approach based on 3D printed models. The target is to visualise the changes in the influential sphere of a country in a more dynamic and interactive manner.

For the purpose of this thesis two prototypes of tangible historical maps were created. They were built from standardised 2D historical maps and feature territorial shifts in x- and y-direction and time on the z-axis. In the course of the work, the impact of today's limited low cost 3D printing techniques on this approach became clearer. Printing is still time-consuming and consumer-level devices have only limited build volumes. Therefore, an application is suggested, which offers users the chance to manipulate certain years in the final 3D model. This results in a custom-defined quantity of detail in the created tangible historical map. The application can enlarge or reduce height in the chosen area and also makes recommendations for certain printer settings based on the user's input. The latter aids the use of this approach for people without background in 3D printing.

In total, this approach proved useful when only a small number of countries with a significant amount of changes over a short period of time is considered.

Contents

CHAPTER 1

Introduction

Although the first attempt of 3D printing[1] dates back to 1983 and the first 3D printer was patented in 1986[1], 3D printing did not become widely adopted until the 2010s. Nowadays, 3D printing is a cost-efficient way to produce a wide variety of customisable parts and pieces. Their purpose ranges from the simplification of fabrication processes to art and education. Futurologist Jeremy Rifkins, senior lecturer at the Wharton School's Executive Education Program at the University of Pennsylvania, even believes that 3D printing signals the beginning of a third industrial revolution.[2] Websites like Thingiverse [3] make printable 3D models globally available to everyone with a computer and access to a 3D printer, which in the future could be nearly every household as prices for printers are dropping. The advantages of reproducibility and easy sharing of 3D models are going to play a great role in many different areas. In this thesis I would like to focus on educational and creative aspects.

History classes in Austria include lessons on the territorial shifts of countries. These changes in the influential sphere are largely due to wars, uprisings or other political reasons. The most commonly used method to teach this matter is aided by huge pull-down maps. The teacher will point at the area where changes have happened and explain them to their students. This mode needs accurate verbal description, as the visual presentation is limited to one static image. I suggest a new approach to the matter, which will aid those who try to describe territorial shifts over time. 3D printing has opened up a viable dynamic and interactive future teaching method, which I am going to describe in my thesis.

It is feasible to create a 3D model of visible changes in a country's border based on pictures extracted from a computer program called "The Centennia Historical Atlas"[4].

[1] `http://3dprint.com/72171/first-3d-printer-chuck-hull/`, accessed 2016-02-08
[2] `http://www.thethirdindustrialrevolution.com/`, accessed 2016-02-08
[3] `http://www.thingiverse.com/`, accessed 2016-02-29
[4] `http://www.historicalatlas.com/`, accessed 2016-11-16

This program provides a simple graphical user interface to navigate through time while showing territorial shifts. There would have been a few different approaches to represent the obtained 2D data in a 3D model. I suggest using the z-axis for time and the x- and y-axis for any loss or growth of territory. When all single slices are stacked this way and correctly aligned, changes in a country's influential sphere are easily observed.
The extracted pictures from "The Centennia Historical Atlas" show movement of any country's border over time, subdivided in 10 slices per year. So a country can be represented by 10 different expansions, which can lead to 10 varying shapes over one year in the final print. As printing all countries or offered time slices would have exceeded my resources I have decided to concentrate on Austria and Germany from the beginning of World War I to the end of World War II, as wars are always points of interest when looking at historical maps. Also, these two wars cover a span of approximately 30 years in total, with a short peaceful period in between. The build volume of consumer-level 3D printers is limited to 15 to 20 centimetres in most cases, so a shorter period of time allows for more height for each slice in the final print. This benefits the printed result by preventing parts, which could easily break while removing supports or when the model is touched too often.

By extending the model in a third dimension, my approach can help visualise the movement over time in a literally tangible way. This might prove very useful to teach such a dynamic matter as historically influential sphere changes. Students would not depend on their imagination only, but could actually flip and turn their printed model to explore different slices from every angle. If they have a 3D printer available, they are able to print their very own customised tangible historical maps with the help of my application, which allows to "zoom in" on specific years. Those years of interest will then be given more height in relation to the other slices and are therefore printed bigger in the final result.

The rest of the thesis is divided into four main parts: In chapter 2 I present some brief insight into related work as well as state-of-the-art 3D printing approaches and compare those to my approach. This overview is followed by a summary of used methods, concepts and technologies in chapter 3. The chapter also includes a more superficial description of my suggested approach, the application and its use cases.

For more details on the architecture and the actual implementation of the approach see chapter 4. There I gathered in-depth information about techniques and programs used to extract, convert and stack the 2D data to a finished 3D model, as well as the laid out functionality and design of the application. The thesis concludes with a discussion of the 3D printed results and some possible solutions to issues I encountered while working on my approach in chapter 5 as well as a summary and an outlook on future work in chapter 6.

CHAPTER 2

Related Work

There is not one but many methods to 3D print an object, all of them being additive manufacturing techniques. Most commonly used is Fused Deposition Modeling (FDM) [1]. Printers using FDM are widespread amongst consumers, as they have become cheaper over the last years. A schematic of a typical FDM printer is shown in figure 2.1. FDM printers melt filament and apply it first to the printers base, later to the already existing structure of the model. For certain areas the printer has to add support structures so that the model does not collapse during the printing process. The cheapest filaments are made of plastic, but there is research into the use of other materials, e.g. ceramics.[2] Some other 3D printing methods are Stereolithography (SLA), Solid Ground Curing (SGC) or Selective Laser Sintering (SLS). SLA and SGC both work with liquid photopolymers which harden when exposed to (UV-)light. SLS is a technique based on powdered material, the loose powder is melted together by very powerful lasers to add layers. In 1995 R. Ippolito and L. Iuliano compared different printing methods in terms of accuracy and surface finish. They found no significant precision differences between the methods but stated that many errors are due to unskilled operators or depend on the material used.[3] For the purpose of this thesis FDM was used as it is the most easily available option. More about the actual printing process can be found in chapter 4.

There are a lot of different use cases for 3D printing, ranging from 3D printing as teaching method [4] to research, which proves 3D printing food [5], as well as 3D printing organs [6] could be possible. As mentioned in the introduction, 3D printing is currently seen as the beginning of a third industrial revolution. In 2012 Berman published a paper listing advantages of 3D printing compared to conventional production techniques.[7]

[1] `http://www.livescience.com/39810-fused-deposition-modeling.html`, accessed 2016-09-28

[2] `https://fused-deposition.wikispaces.com/file/view/fdm_process.jpg`, accessed 2016-09-28

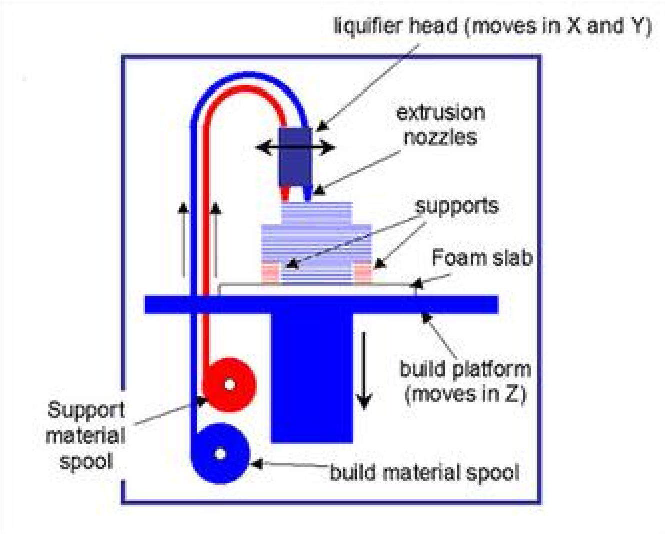

Figure 2.1: Schematic of a FDM printer[2]

The biggest advantage as described by Berman is the constant, rather low cost, which allows a production on demand and thereby does not cause any unsold finished products.

Additionally, the usage of 3D prints is beneficial to different kinds of imaging methods, especially in medical set-ups. Based on data acquired through Computer Tomography (CT) or Magnetic Resonance Imaging (MRI) scans, 3D models can be generated. The data, which is normally sliced into 2D pictures, can now be printed as tangible structures. These prints aid the planning of difficult surgeries, are used in medical education or can provide further insight into not yet understood physiological processes.[8] There is also great potential in printing patient-specific implants, in fact there have already been surgeries where 3D printed parts were used.[3] As Rengier et al. stated in their paper, one of the main limitations of 3D printing is the finite build area of the most commonly used printers. There is only a fixed height and width they are able to cover and so it is not yet possible to produce extremely large pieces.[8]
The same limitation is one of the main reasons to focus on an especially interesting period in history (e.g. wars) when creating tangible maps. Bigger dimensions would allow to cover more time and yet keep a steady amount of detail.

Depending on the quality of the initial images, the results of 3D prints created from 2D data may vary greatly. In 2012 T. Kersten and K. Mechelke compared a 3D model created by laser scan technology to a model created by the stitching of 2D images. Although the camera was a high-end digital camera and the amount of photographs highly exceeded the number of scans, the laser scanning was more efficient in creating an accurate 3D model.[9] Even if the initial images offer enough data for creating a 3D model, the resulting quality will highly depend on the post-processing used. D. Thingdahl and L.

[3] `http://www.bbc.com/news/technology-16907104`, accessed 2016-08-24

Van Gool offer a free-to-use public online tool to create 3D models from pictures taken by consumer cameras.[10] Their focus lies on automatically creating textured and dense models. They start with a 5 step reconstruction mechanism, including Recalibration, where they take stored camera data in account and Epipolar Geometry Computation, where Speeded Up Robust Features (SURF) are found and matched in image pairs. If there are less than two images for an initial calibration matrix, there is an optional Self Calibration step. Afterwards comes Sparse Reconstruction, where a first triangulation of the object takes place. In the last phase, Dense Reconstruction, 3D coordinates for each pixel are calculated. The resulting depth maps are used as input for creating the 3D mesh and therefore, aid the accuracy of the final result.[10]
Historical maps mostly contain 2D information only and make such an approach unnecessary. The maps used for this work feature borders in black and any kind of water in blue, countries are filled in with distinguishably different colours. A simple colour and edge detection is suitable to get all the information needed to extract a country's border from the images used.

When the 3D model is finished there are methods to prove if it is printable or not. Errors occur in connection with solidity, flipped faces, non-manifold geometries or when problems with intersecting objects exist. For simple structures some of these errors can be fixed by certain algorithms,[11] but detailed structures with lots of input vertices tend to get unmanageable. Jamshidi et al. proposed using a SQL based database to speed up said algorithms.[12]
Natural borders create very complex outlines and even with applied smoothing tend to have a high number of anchor points. Using automated repair kits for occurring errors has therefore not proven viable for my approach. That fact, unfortunately, extended the amount of manual data preparation needed.

In April 2016, the Austrian Federal Institute of Geology (Geologische Bundesanstalt (GBA)) showcased a tangible map of Vienna's geologic underground at their "Long Night of Science" exhibition. The 3D print is separable into 4 individual pieces, each representing one or more chronostratigraphic zones underneath Vienna. The final model mapped the underground to a depth of 5.500 meters and the separable pieces were printed in different colours.[4] The multi-colour print helps to distinguish between slices even if the 3D model is put together and seen as a whole.
There are FDM multi-colour printing options through the deposition of more than one filament to the same 3D printed model. Multi-colour printers feature more than one hot nozzle, through which the filament is distributed. J. Hergel and S. Lefebvre wrote a paper in 2014, suggesting improvements to the printer's software as well as changes in the 3D model's printing angle to get a cleaner multi-filament print.[13]
A multi-coloured print might prove beneficial to the result of this work. It would make it easier to distinguish between years on the final print and would help to show enlarged areas of interest when printing customised models.

[4] `https://www.geologie.ac.at/news/news-artikel/article/3d-druck-des-tieferen-geologischen-untergrunds-von-wien/`, accessed 2016-11-16

CHAPTER 3

Methodology

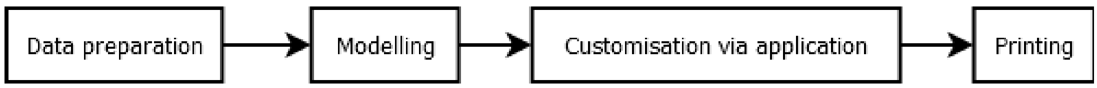

Figure 3.1: Formal pipeline

In this chapter, I am going to discuss used concepts and explain methods and why they are important for the suggested approach. It is structured into four sections and covers the main steps from 2D image data to a final 3D printed model, as shown in figure 3.1. For the actual implementation, refer to chapter 4.

One of the most important tasks was to find a source for historical data to build this work around. Historical maps often lack the option of easy comparison due to the imperfect measuring methods used in their creation. But comparability is a vital necessity for tangible historical maps as proposed in this thesis. I finally found a good source for initial data in "The Centennia Historical Atlas", which already provides the information needed unified on a standardised map of Europe. As the contents of the program are copyright protected, I negotiated a fair use agreement with the creator and got permission to use maps extracted from the atlas for my work.

3.1 Data preparation

The above mentioned program has a built-in dialog box to save the currently viewed map. Unfortunately, there are only a few options to specify output parameters (e.g. resolution, black and white (B/W) or full colour, etc.). Therefore, the extracted data had to undergo some preparation before it was utilisable for 3D modelling. Figure 3.2 gives a more detailed overview of necessary preparation steps. To get the best outcome available, maps were extracted with the highest resolution possible and in full colour.

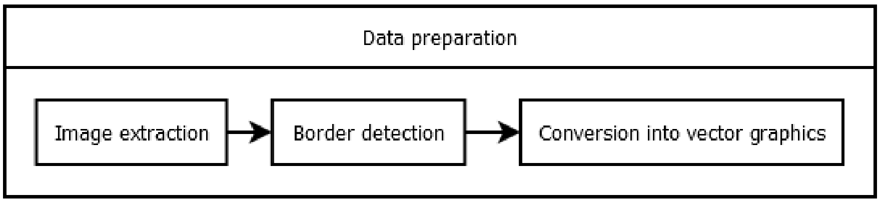

Figure 3.2: Steps of data preparation

The latter was necessary to easily distinguish countries from each other in the following steps. Each of the maps represents one time slice, which means it is an image showing all of Europe at a certain point in time between July 1914 and May 1945. These dates mark the beginning of World War I and the surrender of Nazi Germany in World War II. The decision for this period of time was based on the fact that wars most likely produce changes in the influential sphere of participating countries. Due to the fact that data preparation was not fully automatable and took up a significant amount of time, I concentrated on only two countries which took a major part in both wars - Austria and Germany.[14] [15] As mentioned before, the extraction process was restricted to very few selectable parameters and those did not include the final format of the images. All maps had to be saved in a raster graphics file format, which is not suitable for 3D modelling. The limitations of "The Centennia Historical Atlas" also allowed only for manual image extraction, making it one of the more time consuming tasks.

To create utilisable input for 3D modelling, the border of Austria or Germany had to be detected in each extracted map. If the images only contain borders, they can be vectorised more accurately later. Vectorisation is needed to build a printable 3D model from the obtained data. Border detection was fully automated with MATLAB [1] and part of the automation code can be found in section 4.1. MATLAB offers a variety of pre-implemented image processing tools, including different edge detection methods. Edge detection allows to manipulate images in a way to keep only the outer edges of their content. I used the Canny Edge detector as it produced the best output compared to all other algorithms (e.g. Prewitt operator) offered by MATLAB [2] and the fact that it uses smoothing on the image as well. Smoothing is highly beneficial to this work, as it reduces the details of a country's natural border. Fewer details mean fewer anchor points after vectorisation and therefore fewer vertices in the 3D model. Canny proposed his edge detector in 1986 [16], and it is still one of the best known algorithms for this purpose. The detector uses 5 main steps to create a final result: a Gaussian filter for smoothing, determination of the intensity gradient of the image, non-maximum suppression for less false edges, application of a double threshold for noise reduction and finally tracking edges by hysteresis to suppress weak edges.

[1] `https://mathworks.com/products/matlab.html`, accessed 2017-01-04
[2] `https://mathworks.com/help/images/ref/edge.html`, accessed 2017-01-04

The need to vectorise the resulting B/W images very accurately and mostly without manual intervention led to the decision to use a closed-source software. Attempts with open-source software (e.g. Inkscape[3]) always failed one of the above mentioned criteria. Adobe Illustrator CC's[4] image trace tool offers vectorisation with just a few clicks and the option for different parameter presets. The "Silhouette" preset proved to be suitable to create (mostly) closed, accurate paths. If paths remained open after tracing they had to be closed manually in order to avoid them causing any problems in the modelling process. As the used software is closed-source, no specifics can be given on the actual implementation of the vectorisation process. Some of the most commonly used vectorisation techniques were summarised by L. Wenyin and D. Dori in 1999. They classified methods in six categories: Hough transform-based, thinning-based, contour-based, run-graph-based, mesh-pattern-based, and sparse-pixel-based. With the exception of Hough transform-based algorithms each of them builds around the following three steps: determination of important points on the medial axis, line tracking and line segment approximation or polygonalisation.[17]

If the original map features islands or parts, which were never directly attached to the country's main territory, they have to be removed. Otherwise, the best case scenario would be that they get printed on support structures somewhere beside the actual model. In the worst case, they would render the created model unprintable. I decided to remove these parts after vectorisation because connected paths are more easily deleted from a vector graphic than structures from a raster graphic.
After vectorisation all criteria for 3D modelling were finally met.

3.2 Modelling

As mentioned in the introduction, I decided to use the z-axis for the representation of time. This decision was influenced by the fact that the extracted data already provided x- and y-coordinates. Based on that, the modelling environment had to offer two things: simple importation of the prepared paths and an easy way to extrude them in z-direction. At best, it is also able to fill in the paths to create a solid object. As there are sometimes gaps in the influential sphere, which create holes in the map, this has to happen in an intelligent manner. I found a good software option in OpenSCAD[5], an open-source 3D modelling scripting language. It offers a direct import of paths from certain vector graphic formats, an extrusion operation and automatically fills in the area between paths when such are read in.

All steps performed on an imported map in OpenSCAD before the desired 3D model is finished are shown in figure 3.3. The height for extrusion in z-direction is set later by the suggested application. This approach creates the possibility of customising the final model. Customisation will be discussed in the next section and details on the

[3] `https://inkscape.org/`, accessed 2017-01-04
[4] `http://www.adobe.com/products/illustrator.html`, accessed 2017-01-04
[5] `http://www.openscad.org/`, accessed 2017-01-05

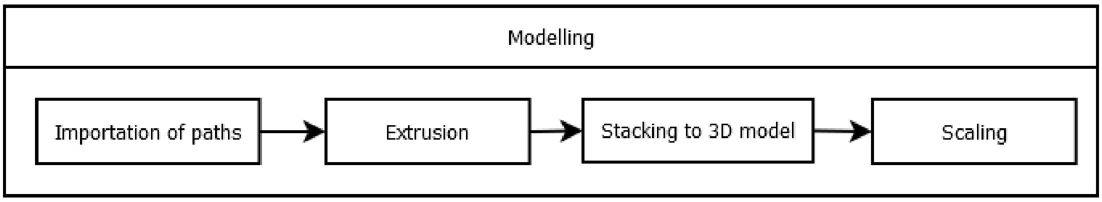

Figure 3.3: Steps of modelling

implementation of the application can be found in section 4.3. The last step, scaling of the x- and y-input, is triggered by the high resolution of the initial images extracted from "The Centennia Historical Atlas". High resolution creates greater relative distances in between anchor points when the image is vectorised and therefore, creates very large input coordinates for the modelling language. Without scaling the resulting model would exceed the build volume of most 3D printers, while scaling too much would render some of the more fragile parts unprintable. More details on the script for the 3D models of Austria and Germany can be found in section 4.2.

3.3 Customisation via application

Due to the finite printable height only a certain amount of detail can be displayed on the z-axis. Also, historical maps might not differ greatly over some periods, therefore will not result in very interesting tangible maps for these times. Due to this fact and a fruitless search for existing software offering the option to manipulate selected parts of an already existing 3D model in z-direction, I propose a new application. The application is interfering with the height of each slice in the printable result of the previous modelling process. The suggested approach works as follows: The user can choose a year as the centre of an area of interest, the chosen year will then get more room on the z-axis. Slices surrounding the centre will be affected as well, but to a lesser degree.
The possibility to choose more than one area of interest was discarded as each slice has to maintain a certain height to stay printable. Models with two or more areas of interest struggled to display more details than the original model with constant height. However, I implemented the option to reduce height on a less interesting part, which then creates more space for the other parts automatically. Even if no manipulation of the height is needed, users have to specify certain parameters (e.g. resolution) to create the best possible outcome for their own 3D printers.

I decided against offering the user an option to change the maps in x- and y-direction as well. As mentioned in the previous section scaling was applied to the map because of the size the imported data had. As Germany once nearly conquered all of Europe its map is quite large in certain parts. Scaled to 20 percent it measures a little over 14 centimetre in x-direction in some years, with most consumer-level 3D printers having a build volume of 15 centimetre in each direction. Reducing the detail in x- and y-direction any further would have resulted in structures prone to breaking when all supports are removed. Using a scale greater than 20 percent increases the needed filament (when FDM printer is used) and printing time in an impractical way. Therefore, I saw no reason to include a customisation option for the x- and y-scaling parameters.

The application was implemented in C++ wit Qt Creator[6]. Qt Creator has an embedded Graphical User Interface (GUI) designer and allows for easy integration of new widgets. The need for a new tool was aroused by my plan to visualise the arrangement of each slice on the z-axis for a more intuitive user experience. But there was no widget to be found, which offered the kind of visualisation I wanted. See section 4.3 for details on the newly created software.

3.4 Printing

After all parameters for the 3D model are set, printing is the final step to a tangible historical map. An overview of the printing process is shown in figure 3.4.

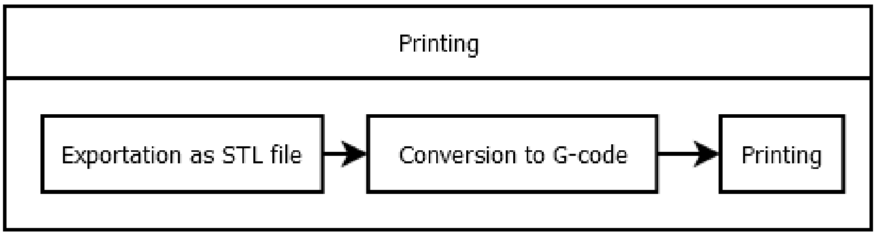

Figure 3.4: Steps of printing

After rendering, the resulting model can be saved in STereoLithography (STL) format, which is still not understood by 3D printers. STL files are widely used for rapid prototyping, 3D printing [18, p. 237] and on sharing platforms like Thingiverse.[7] For the actual printing the file must be converted to G-code, either via open-source software, such as SLICER[8] or a printer's own software. Anazalone et al. describe a similar process in their paper from 2013: "The open-source firmware (Repetier-firmware) translates G-code commands into pulses to the stepper motors, controlling the motion of the stage. (...) Models based upon the stereolithography (STL) standard are converted to G-code by software colloquially known as a "slicer", which creates patterns from uniformly thick slices of the model through the z-axis."[19] The actual printing might take several hours depending on the intended height of the print as well as on the printer settings. I printed the results on a consumer-level printer to show the feasibility of a low-cost approach suitable for educational purposes. Details on the used printer and settings can be found in section 4.4.

[6] https://www.qt.io/ide/, accessed 2017-01-11
[7] http://www.thingiverse.com/, accessed 2017-01-07
[8] http://slic3r.org/, accessed 2017-01-01

The model should be printed with the highest infill setting possible because some time slices have great territorial expansion. These parts of the final model would get very filigree and could break easily, especially when the printed supports are removed. I also suggest using a layer height compatible with the height of each slice. If the height of each slice is divisible by two, the recommendation is a layer height of 200 microns, if divisible by three, 300 microns are fine enough.

CHAPTER 4

Implementation

This chapter covers the suggested implementation of the already introduced steps from chapter 3. Before going into more details, the architecture of the approach is briefly summarised. A graphical overview is shown in figure 4.1. The upper part of the chart covers data preparation (section 4.1); the first item in the second row belongs to modelling (section 4.2). Customisation (section 4.3) is represented in the second item in the second row, the last two items are part of the printing process (section 4.4). The conversion to AutoCAD Drawing Exchange Format (DXF), which is the last step of data preparation, could potentially be omitted due to the fact that the newest development snapshot of OpenSCAD (2016.11.21/2016.10.04) is able to read Scalable Vector Graphics (SVG) files directly. All steps shown in figure 4.1 will be explained in the following sections.

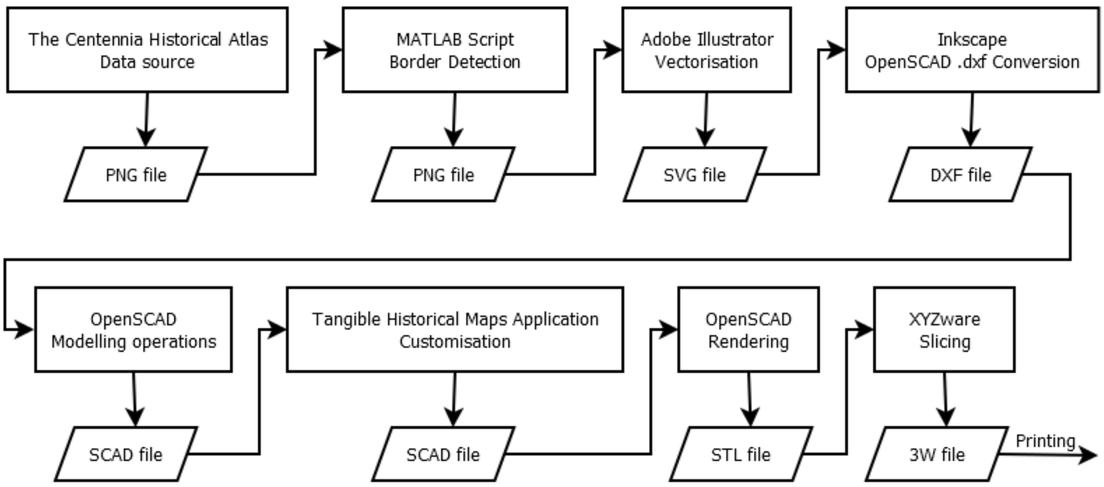

Figure 4.1: Architecture of this approach

4.1 Data preparation

As mentioned before, data preparation can be broken down into three steps: Extraction, border detection and vectorisation. The result of each step is shown in figure 4.2.

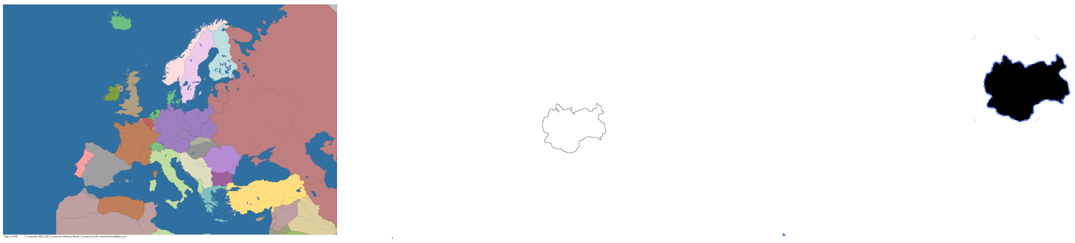

Figure 4.2: Map after extraction, after border detection and after vectorisation

The initial data for this work was extracted from “The Centennia Historical Atlas” (version 3.11). The program offers 10 time slices a year, making it a total of 320 time slices from the beginning of World War I to the final occupation of Germany at the end of World War II. To avoid unnecessary repetition, only those slices with changes in the influential sphere of the targeted country were extracted. The images were extracted in colour (RGB (red, green and blue) space), set to the highest resolution possible and saved as Portable Network Graphics (PNG) files.

Algorithm 4.1: MATLAB code snippet for border detection

```
a. red = image(:,:,1);
b. r_logic = red == REDVALUE ;
c. green = image(:,:,2);
d. g_logic = green == GREENVALUE;
e. blue = image(:,:,3);
f. b_logic = blue == BLUEVALUE;
g. logic = r_logic & g_logic & b_logic;
h.
i. canny = edge(logic,'Canny',0.5,10);
j. canny = 1-BW;
```

The PNG files were then fed to the automation script for border detection, of which a part is shown in algorithm 4.1. In lines a to g, I implemented a simple logical colour filter. The placeholders “redvalue”, “greenvalue” and “bluevalue” (line b, d and f) must be replaced with the RGB values of the targeted country, e.g. 159, 127 and 191 for Germany. Combining the separately detected colours results in a single B/W image containing the country’s shape in white on a black background. B/W images work best with the following edge detection due to the high contrast between country and background. The motivation for using Canny edge detection is discussed in section 3.1. The edge detector is pre-implemented in MATLAB and can be called with “edge(input, ’Canny’, threshold, sigma)". I used the edge detector with a threshold of 0.5 to reduce minor noise in the image (e.g. lines created by rivers flowing through a country) and sigma 10 to smooth out the image just enough to close most gaps before detection. Figure 4.3 shows a comparison of images produced with threshold: 0.4 / sigma: 10, threshold: 0.5 / sigma: 10 and threshold: 0.5 / sigma: 9. Both settings deviating from the proposed above create noise in the middle of the map of Austria-Hungary where Lake Balaton is located.

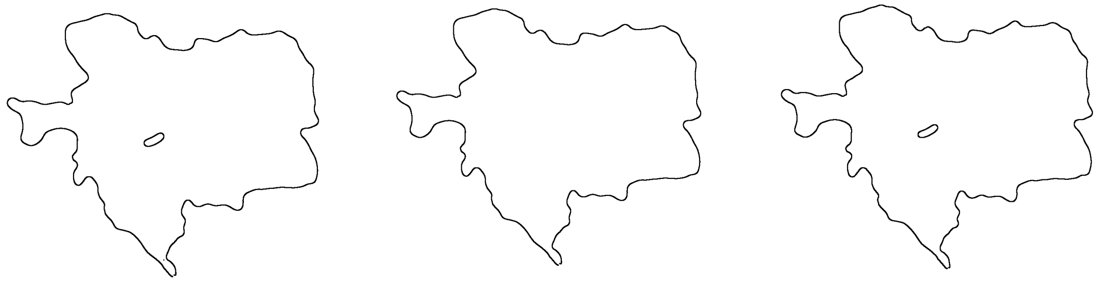

Figure 4.3: Comparison of edge() settings, left and right image show unwanted noise

The B/W image with the resulting edges is first reversed (line j) and then the edges are being thickened to close still existing gaps. The script also creates a small black square in the lower left corner, which is useful for stacking the maps more easily (see section 4.2 for details). When border detection is finished the maps get stored as PNG files again.

For vectorisation I used Adobe Illustrator CC. The actual vectorisation process can be done by two clicks after opening the file and selecting everything in the picture. The first click chooses the preset - as mentioned before the pre-set I used was "Silhouette" - the second click is pressing the convert-button. In some cases minor issues had to be fixed manually after conversion, most of the time these issues were caused by open paths. After closing all gaps I deleted the inner circle of each path, so OpenSCAD would fill in the whole model later. If the inner part of an outline is still intact when the paths are fed to OpenSCAD, it will only render the outlines of this part. Afterwards each time slice was saved in SVG format.

As mentioned before, OpenSCAD does feature direct import of SVG files in its newest development snapshot, but had not yet done so when I worked on my model. Therefore, a last conversion was necessary before the data was ready to be imported in OpenSCAD. To import 2D data it had to be in DXF format without splines[1], since splines are not supported by OpenSCAD. There is a plugin to create such DXF files[2] for Inkscape (version 0.91). Unfortunately, Adobe Illustrator CC saves SVG files without certain parameters (e.g. svgwidth) needed by the Inkscape plugin. I obtained those parameters by opening and re-saving all time slices in Adobe Illustrator CS3, with which I had

[1] `http://www.autodesk.com/techpubs/autocad/acad2000/dxf/spline_dxf_06.htm`, accessed 2017-01-07

[2] `https://github.com/brad/Inkscape-OpenSCAD-DXF-Export`, accessed 2017-01-07

already successfully exported and converted SVG to DXF before. Then I opened all of them in Inkscape and saved them via the normal “Saving as”-dialog box as “OpenSCAD DXF Output”, which is provided by the before mentioned plugin.

4.2 Modelling

All modelling steps were entirely done in OpenSCAD (version 2015.03-2). At first the prepared paths had to be imported. Each imported DXF file already contained x- and y directions, z-directions were created via extrusion. The height of each slice is determined by the planned height of the final model and is filled in later when customisation is applied. An example import and extrusion can be found in algorithm 4.2. Example height “a” (used in line c and f) as well as height “af” (line x) will be declared later through the proposed application. To create a model out of the accumulated slices, each slice has to be placed on top of the previous one. To prevent gaps and allow the union()-operation to work, each slice is placed slightly overlapping with its predecessor. Therefore, the translation of a new slice is calculated by the gathered height of all predecessors minus an overlap, which should be restricted to a bare minimum. The stacking was eased by the small black square created in the lower left corner of each image in the data preparation process. OpenSCAD uses the lower left corner of imported files for alignment, through the little structure there all slices were automatically aligned.

Algorithm 4.2: Code snippet from the 3D model’s script

```
  ...
a. scale ([0.2,0.2,1])
b. union() {
c.      linear_extrude(height = a)
d.              import("1914_5.dxf");
e.      translate([0,0,d1]){
f.              linear_extrude(height = a)
g.                  import("1914_6.dxf");}
  ...
w.      translate([0,0,d76]){
x.          linear_extrude(height = af)
y.                  import("1945_4.dxf");}
z. }
```

The stacked slices are then joined via the union()-operation and scaled to 20 percent in x- and y-direction. The reasons for scaling have been already mentioned in section 3.2. Due to the fact that I have extracted different slices for Austria and Germany, depending on where territorial shifts happened, I had to create two different scripts. If no slice contains more than one tenth of a year, one script for both countries is feasible with a simple change of underlying data.

4.3 Customisation via application

The proposed application offers users the chance to customise the created 3D models via a simple GUI (see figure 4.4). The icons used in the GUI are part of the “Default Icon” library by Apostolos Paschalidis/interactivemania[3] and were released under Creative Commons (CC) license.

The application offers the following functionality: The centre for either increased or decreased height can be selected by first clicking on the period of interest. When a period is chosen, another menu appears from which the year of interest is picked. The user can also set the 3D printer’s minimal layer height, so a layer height warning is shown when the height of a slice is below the set minima. If only one model should be printed, Austria can use more space on the z-axis per slice, because its model ends with 1938 as Austria became part of Germany in 1938. If a matching set should be printed Austria is limited to the same height per slice as Germany, the result will be comparable when printed. All settings influence the multi-coloured bar at the bottom of the GUI (see figure 4.4). It visualises all years and their thickness in the final print and the relation to the maximum printing height. Underneath the user can find information about the actual height of the print, the height of the smallest and the biggest slice as well as the recommended layer height for printing. These pieces of information are calculated from the given parameters. If a year is selected, the changes, which are applied to the bar are applied to the model as well. The user can also choose between instantly rendering an STL file or saving the model(s) first in OpenSCAD file format (SCAD) format to render them manually later. The increase of height for the year of interest is done by applying a

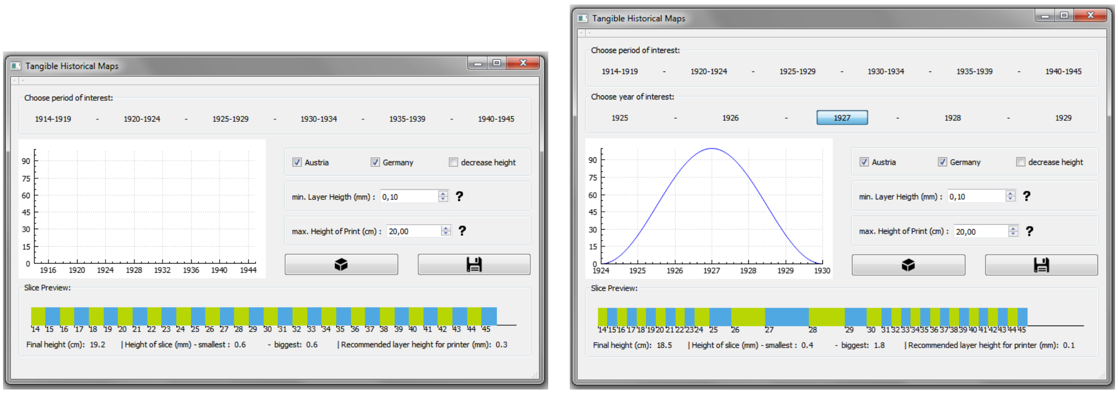

Figure 4.4: GUI on start-up and with 1927 as centre of interest

sine function. This sine function is used to get smoother result and can be seen in the plot on the left side of the application. It is only visible when a year is chosen and also visualises the influence of the “zooming” on the slices surrounding the centre year. The plot makes use of the open-source QCustomPlot-widget[4].

[3] www.defaulticons.com

[4] http://www.qcustomplot.com/, accessed 2017-01-12

Algorithm 4.3: Qt/C++ code to increase the height

```
a. for(int j = 0; j <= 6; j++){
b.  double sine = sin((-M_PI/2)+j*(M_PI/3));
c.  double sinePlusOne = sine+1;
d.  double slice = floor((calcHeight+
                     (calcHeight*sinePlusOne))*10)/10;
e. }
```

The code for calculating the increased height is shown in algorithm 4.3. The variable "calcHeight" is determined by dividing the maximum height by the number of slices and truncated to tenths digit. The peak of the sine function is applied to the centre of interest, the surrounding two years in each direction are increased to a lesser degree. The third year in each direction is not affected as the calculated sine function is zero there. As one period of the sine produces results in between -1 and 1, 1 is added in line c to only ensure only positive numbers are used for multiplication later. The results of line d are the final heights of the enlarged slices, they are also truncated to tenths digit. To calculate the height of all other slices, the height of the increased part is summed up and subtracted from the maximum height. The result is divided by the number of remaining slices and truncated to tenths digit again.

Algorithm 4.4: Qt/C++ code to decrease the height

```
a. for(int j = 0; j < 11; j++){
b.  double sine = sin((-M_PI/2)+j*(M_PI/5));
c.  double sinePlusOne = sine+1;
d.  double slice = floor((calcHeight*sinePlusOne)*10)/10;
e.  decreaseArray[j] = slice;
f. }
g. double min = decreaseArray[5];
h. double max = decreaseArray[0];
i. for(int i = 0; i < 11; i++){
j.  double tmp = getEqualizedHeight(decreaseArray[i],
                            min, max, minLayer, calcHeight);
k.  double slice = floor(tmp*10)/10;
l.  decreaseArray[i] = slice;
m. }
```

For decreasing the height I used the same approach as for increasing, but had to alter it according to printer limitations. The sine function was calculated as before, this time with a radius of five years around the centre of interest. Using a five year radius allows for more height to be shifted to the other slices later. Then I calculated temporary factors (line d) which then were equalised to the range of "minLayer" (minimal layer height) to "calcHeight"(line j). The rest of the calculations were done in the same way as for increasing the height. For my prototype I used constant 0.3 millimetre slice height on both models.

4.4 Printing

Rendering the model for printing can be triggered from inside the application or done manually, opening the SCAD (OpenSCAD file format) file and pressing F6 in OpenSCAD. The built-in option also uses the OpenSCAD renderer. Regardless of the rendering process, the only saving option is STL. As I used the XYZprinting da Vinci Mini 3D printer[5], I was obliged to slice my model with the printer's own software (XYZware). I used 300 microns layer height, solid (90%) rectangular infill, normal speed, normal shell thickness and supports. The printing of Austria took about six hours and eight metre of 1.75 millimetre PLA (Polylactic acid) filament. The estimated printing time for Germany was reduced from 21 hours to eleven hours by splitting it in two parts, the 3D print used about 45 metres of 1.75 millimetre PLA filament.

[5] `http://us.xyzprinting.com/us_en/Product/da-Vinci-Mini`, accessed 2017-01-07

CHAPTER 5

Results and discussion

This chapter contains the discussion of the 3D printed results as well as possible solutions to issues I encountered while working on my approach. It also gives insight into the results of my work. To my knowledge a similar approach to the one suggested does not exist, therefore a direct comparison to related work is not possible.

While implementing the application suggested in this work, I realised a lot of dependencies and special cases, which are in fact avoidable. E.g. if the underlying data started at the first slice of 1914 instead of the fourth (which is where World War I started), one special case could have been immediately eliminated. On the other hand, this would have narrowed the already tight space on the z-axis even more and resulted in loss of detail.

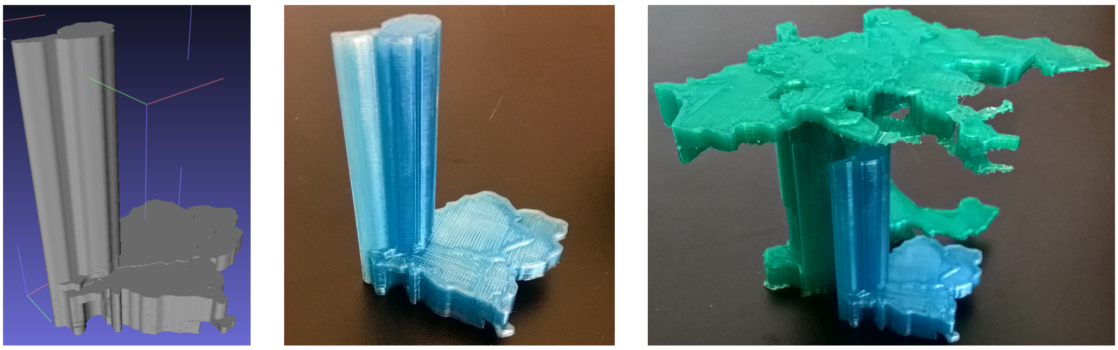

Figure 5.1: Model and prototype of Austria & the latter in comparison to Germany

In figures 5.1 and 5.2 the original 3D models and the printed prototypes are shown. They prove my approach to be feasible, even on a consumer-level 3D printer. Austria has turned out very solid and would even be printable with a smaller layer height than 0.3 millimetres. Germany features some very fragile parts, which barely withstood the removal of the support structure from the prototype. Due to this fact I would implement a warning in the application if the height of these specific slices drops beneath 0.3 millimetre. Especially as this can happen without the user's intent when other slices of the model are enlarged or a nearby slice is the centre for height reduction.

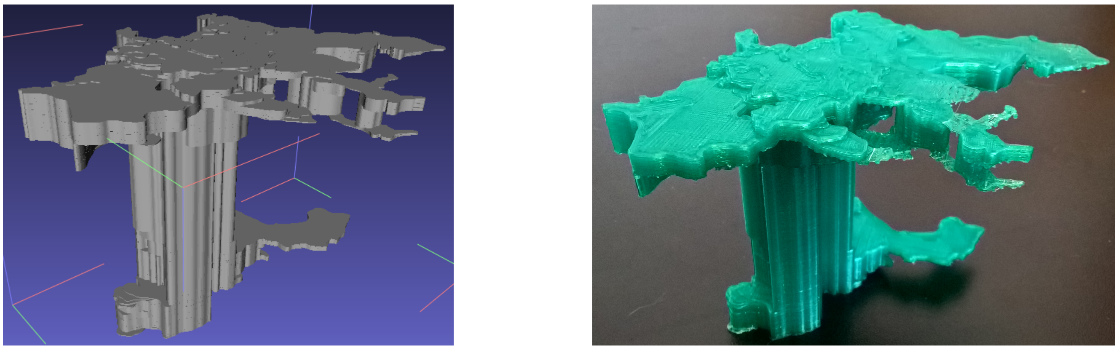

Figure 5.2: Model and prototype of Germany

Austria and Germany do not fit together like puzzle pieces as they share the same border and I did not create any separation in the data preparation process. This work never intended them to do so, but if puzzle-like behaviour is desired, Austria should be scaled to approximately 19 percent in x- and y-direction instead of 20 percent. This can be done manually by saving Austria as SCAD and altering it in the created script. The line should look like line a in algorithm 4.2 and should be altered to "scale ([0.19,0.19,1])".

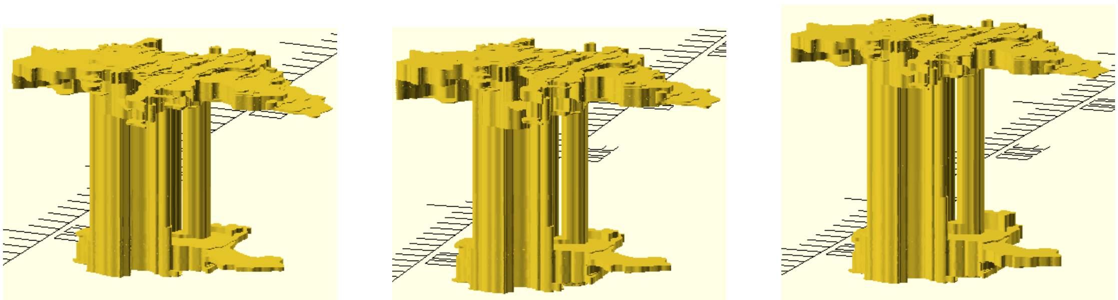

Figure 5.3: Germany: constant height, decreased middle part and increased lower part

The left image of figure 5.3 shows the model of Germany at 0.3 millimetre constant height, same as the printed prototype. The 3D model was created with the application suggested in this work. The application was given the following parameters: 12.75 centimetre maximum height, 0.1 millimetre minimal layer height and constant height. The resulting model uses only 9.6 centimetres of the maximum height, due to the fact that the application has to break the height in same-sized parts and can not violate the maximum height when stacking those parts on the z-axis.
As the printing time of 20 hours per matching set of 3D printed maps is rather long, I strongly suggest decreasing the height on the peaceful period (1918-1938). The centre should be chosen at 1928 to achieve the greatest impact. When keeping the height of the

other slices intact and therefore reducing the overall height, the printing time will be shortened significantly. Otherwise, slices with more expansion in x- and y-direction will become enlarged and lengthen the printing time. The latter can be seen in the middle picture of figure 5.3. The parameters of the first model were kept, but 1928 was chosen as centre of interest for a decrease operation. Due to the unused part of the maximum height, the application now has the chance to expand all non-decreased areas to 0.4 millimetres. Therefore the model has a final height of 10.7 centimetres and uses more area on the z-axis than before the decrease operation.
The right picture of figure 5.3 shows the model with an increased lower part. The chosen centre of interest is 1919, all other parameters where kept again. The final height of the model increases to 11.8 centimetres, which still is within the 12.75 centimetres maximum height. The increase is especially visible on the part which is expanding to the lower right of the image, it is definably thickened by the applied operation.

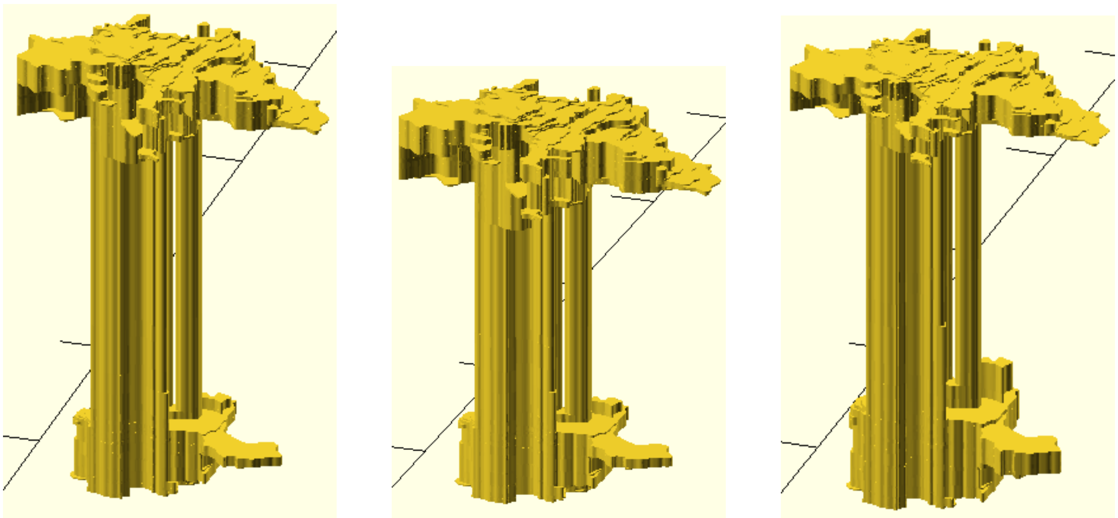

Figure 5.4: Similar comparison as figure 5.3

Figure 5.4 shows similar results as figure 5.3. This time the constant height for the left image was 0.7 millimetres and resulted in a model with 22.4 centimetres final height. This time the 0.7 millimetres were kept for the non-decreased slices, which resulted in a final height reduction of 3.8 centimetres and a final height of 18.6 centimetres (middle picture). 1928 was chosen as centre of interest again to create comparable results. It is clearly visible that there would be more potential for reduction. Unfortunately most consumer-level 3D printer can not print slices thinner than 0.1 millimetres, so to keep the final model printable, the thinnest slices must at least have the minimal layer height set by the user.
In the right picture 1919 was chosen as centre of interest for the increase operation, the resulting model has a height of 22.4 centimetres. The centre slice increased to 2.1 millimetres and the non-increased slices have a thickness of 0.5 millimetres.

A multi-coloured approach would have helped to distinguish between different years and if my model was "zoomed", made this parts visible in the prototype. Unfortunately, multi-coloured prints are costly and there are only a few 3D printing companies worldwide offering them. To provide more insight into the dimensions of x- and y-expansion of a country, a base plate with the outlines of Europe could be produced. When the model is put on top of it, a simpler geographical comparison of the territorial shifts would be possible.

Even with all their limitations and minor insufficiencies the prototypes provide a tangible overview over territorial shifts in between 1914 and 1938. Therefore, they fulfil the initial goal.

CHAPTER 6

Summary and future work

I have described an approach to tangible historical maps in an educational as well as creative context. The suggested design attempts to give more insight into territorial shifts over time than today's teaching methods. The realisation should have been highly customisable and particularly suitable for people without background in 3D printing. The second criterion was met by the proposed application. However, not all of the necessary steps from map to printed model were automatable and the underlying data is not as interchangeable as initially planned. At this time the limitation to a small number of countries with a significant amount of changes over a short period of time appears to be the best approach. The resulting tangible historical maps should be sufficiently detailed and interesting as well as sturdy enough to be used by several students without breaking.

The work also showed, that the created models are printable on a low cost device in an acceptable amount of time. The suggested application offers users some customisation options, but is limited by today's 3D printing technology. With falling prices for constantly improving 3D printers featuring larger build areas, I suggest implementing a customisable radius around the centre of interest in the future. Also, the idea of offering scaling in x- and y- direction should be revised by then.
To create more insight, models could be split in years and a guiding rod with printed-in scale could be used to arrange them in order again. This rod would also aid the visualisation of time on the z-axis and could be of use to make detached areas, such as islands, printable. If the rods are connected to a base plate and getting smaller towards bigger z-axis values, a certain detached part of a country can be placed on the rod and would only slide down as much as the size of the hole for the rod would allow. Therefore it is aligned in time and if the base plate also represents e.g. Europe, it would be aligned geographically correct as well.

In total, this thesis presented a feasible alternative to huge pull down maps or any other method used to teach historical territorial shift nowadays and the approach proved to be viable even for custom-level hardware.

List of Algorithms

Acronyms

B/W black and white. 7, 9, 14, 15

CC Creative Commons. 17

CT Computer Tomography. 4

DXF AutoCAD Drawing Exchange Format. 13, 15, 16

FDM Fused Deposition Modeling. 3–5, 11

GBA Geologische Bundesanstalt. 5

GUI Graphical User Interface. 11, 17

MRI Magnetic Resonance Imaging. 4

PLA Polylactic acid. 19

PNG Portable Network Graphics. 14, 15

RGB red, green and blue. 14

SCAD OpenSCAD file format. 17, 19, 22

SGC Solid Ground Curing. 3

SLA Stereolithography. 3

SLS Selective Laser Sintering. 3

STL STereoLithography. 11, 17, 19

SURF Speeded Up Robust Features. 5

SVG Scalable Vector Graphics. 13, 15, 16

Bibliography

[1] C. Hull, "Apparatus for production of three-dimensional objects by stereolithography," Mar. 11 1986. US Patent 4,575,330.

[2] A. Bellini, L. Shor, and S. I. Guceri, "New developments in fused deposition modeling of ceramics," *Rapid Prototyping Journal*, vol. 11, no. 4, pp. 214–220, 2005.

[3] R. Ippolito, L. Iuliano, and A. Gatto, "Benchmarking of rapid prototyping techniques in terms of dimensional accuracy and surface finish," *CIRP Annals-Manufacturing Technology*, vol. 44, no. 1, pp. 157–160, 1995.

[4] V. Kostakis, V. Niaros, and C. Giotitsas, "Open source 3d printing as a means of learning: An educational experiment in two high schools in greece," *Telematics and informatics*, vol. 32, no. 1, pp. 118–128, 2015.

[5] J. Sun, W. Zhou, D. Huang, J. Y. Fuh, and G. S. Hong, "An overview of 3d printing technologies for food fabrication," *Food and Bioprocess Technology*, vol. 8, no. 8, pp. 1605–1615, 2015.

[6] V. Mironov, R. P. Visconti, V. Kasyanov, G. Forgacs, C. J. Drake, and R. R. Markwald, "Organ printing: tissue spheroids as building blocks," *Biomaterials*, vol. 30, no. 12, pp. 2164–2174, 2009.

[7] B. Berman, "3-d printing: The new industrial revolution," *Business horizons*, vol. 55, no. 2, pp. 155–162, 2012.

[8] F. Rengier, A. Mehndiratta, H. von Tengg-Kobligk, C. M. Zechmann, R. Unterhinninghofen, H.-U. Kauczor, and F. L. Giesel, "3d printing based on imaging data: review of medical applications," *International journal of computer assisted radiology and surgery*, vol. 5, no. 4, pp. 335–341, 2010.

[9] T. Kersten and K. Mechelke, "Fort al zubarah in katar–3d-modell aus scanner-und bilddaten im vergleich," *AVN-Allgemeine Vermessungs-Nachrichten*, vol. 2, no. 2013, pp. 50–58, 2013.

[10] D. Tingdahl and L. V. Gool, "A public system for image based 3D model generation," in *Lecture Notes in Computer Science*, vol. 6930/2011 of *Computer Vision/Computer*

Graphics Collaboration Techniques 5th International Conference, MIRAGE 2011, pp. 262–273, SpringerLink, 2011.

[11] H. Chao, Y. Li, and Z. Ying-ying, "Research on repair algorithms for hole and cracks errors of stl models," in *International Conference on Information and Management Engineering*, pp. 42–47, Springer, 2011.

[12] P. Jamshidi, M. Haddad, and S. Mansour, "A new database approach to improve stl files correction algorithms," *18th International Conference on Production Research*, 2005.

[13] J. Hergel and S. Lefebvre, "Clean color: Improving multi-filament 3d prints," in *Computer Graphics Forum*, vol. 33, pp. 469–478, Wiley Online Library, 2014.

[14] H. H. Herwig, *The First World War: Germany and Austria-Hungary 1914-1918*. Bloomsbury Publishing, 2014.

[15] B. J. Wendt, "Deutschland im zweiten weltkrieg," in *Das nationalsozialistische Deutschland*, pp. 130–190, Springer, 2000.

[16] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[17] L. Wenyin and D. Dori, "From raster to vectors: Extracting visual information from line drawings," *Pattern Analysis & Applications*, vol. 2, no. 1, pp. 10–21, 1999.

[18] C. K. Chua and K. F. Leong, *Rapid prototyping: principles and applications*, vol. 1. World Scientific, 2003.

[19] G. C. Anzalone, C. Zhang, B. Wijnen, P. G. Sanders, and J. M. Pearce, "A low-cost open-source metal 3-d printer," *IEEE Access*, vol. 1, pp. 803–810, 2013.