FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Walk line drawing

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Computer Science

by

Pablo Balduz
Registration Number 1635296

to the Faculty of Informatics
at the TU Wien
Advisor: Ivan Viola

Vienna, 25th April, 2017

Pablo Balduz	Ivan Viola

Erklärung zur Verfassung der Arbeit

Pablo Balduz
Medwedweg 3, 1110 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, 25. April 2017

Pablo Balduz

Acknowledgements

First of all, I would like to express my sincere gratitude to my advisor Ivan Viola for giving me the opportunity to work on this Bachelor thesis at the Institute of Computer Graphics and Algorithms from Technische Universität Wien. I really appreciate all the counsel given in our weekly meetings, as well as his predisposition to help in any aspect related to the project.

Furthermore, I am really thankful to my friends and family for their support during all my studies and especially during my stay in Vienna working on this thesis.

Kurzfassung

In the recent years, consequence of technology improvements, a new kind of art has appeared. It is called GPS art and consists in drawing on a digital map by recording the path followed using a GPS device.

The fact is that not everyone is able to make the drawing of a certain figure. Just the so-called GPS artists come up with the path, so it makes this kind of art not reachable to some people.

The idea of this thesis is to enable people to create GPS art without relying on imagination to come up with a path for a certain figure. In order to achieve that, a system that finds that path in a map for a given figure as input has been developed. In order for people to use this system, it has been integrated within a mobile application, so users are able to find the path to follow easily.

Abstract

In the recent years, consequence of technology improvements, a new kind of art has appeared. It is called GPS art and consists in drawing on a digital map by recording the path followed using a GPS device.

The fact is that not everyone is able to make the drawing of a certain figure. Just the so-called GPS artists come up with the path, so it makes this kind of art not reachable to some people.

The idea of this thesis is to enable people to create GPS art without relying on imagination to come up with a path for a certain figure. In order to achieve that, a system that finds that path in a map for a given figure as input has been developed. In order for people to use this system, it has been integrated within a mobile application, so users are able to find the path to follow easily.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Global Positioning System (GPS) technology provides geolocation and time information to a receiver anywhere on the Earth where there is a clear line of sight to four or more GPS satellites. It provides positioning capabilities to military, civil and commercial users. In daily life we usually use it to find a route to a certain destination or to know where we are, but some people have experimented with other uses of this technology. It is also used to create art.

GPS arts, also known as GPS drawing is a method of drawing that uses GPS technology to create large-scale artwork. It combines art, movement, and technology. Tracks of a journey can automatically be recorded into the GPS receiver's memory and can subsequently be downloaded onto a computer or visualized on a smartphone for example.

People who create this art are called GPS artists and they follow two steps in order to get their drawings. They first choose a running app that can track their path during the journey and then, they make an outline of what they want to draw. Some artists approach is to use a run-mapping tool like mapmyrun.com to sketch out the exact route online. Once having the route drawn up, artists take a screen shot, which they then check during the run to make sure they have not gone off course.

Nowadays everyone is very close to a GPS device, everyone has a smartphone and all of them provide this technology. The main idea of this project is to bring this kind of art close to everyone but not just the artists that first arrange the drawings in their heads. That is to make the process of finding a concrete drawing in a map much more automatic and easy.

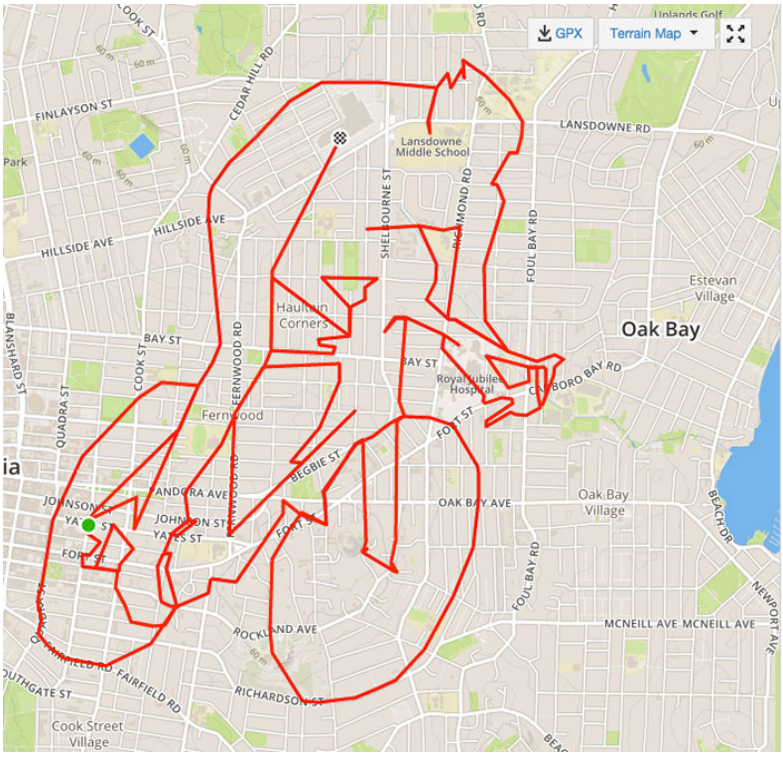


Figure 1.1: Example of GPS art. [1]

1.2 Methodological approach

The aim of this project is to enable people to create GPS art in an effortless way by developing a system of technologies that take as input a line drawing and searches in a chosen map the candidate areas to place the figure in a pedestrian accessible area. The system develop should place a drawing in a digital map of choice. This drawing will be given as input of the system and then it will search for the best areas in which the drawing is more likely to be represented. The area reached by the used map will limit the drawing placement. This whole system will be implemented on a mobile platform. The reason why is the final goal of this kind of art. Once having an outline of a drawing, the next step is to go outside to follow the path while recording it and that is something that mobile platform enables but a desktop application does not. In order to reach most of smartphone users, it is been decided that the app will be developed for the Android platform due to the majority of users it has.

1.3 Structure of work

A revision of the technologies used in this project is presented in Chapter 2, as well as some current related applications that have been the motivation for this project. In

Chapter 3, everything related to the developed system and how it works is explained in detail. Regarding the implementation part, an explanation of the technologies and work environments is made in Chapter 4. Finally, a summary and an outlook for future work in chapter 5 conclude the thesis.

CHAPTER 2

State of the art

This chapter makes an overview on the current technologies used in the making of GPS arts, and also on the technologies involved to achieve the project goal, which is making the whole process of creating this art effortless. In the first part of this chapter a review of the current applications used in the creation of GPS arts is made and in the second part an analysis of the technologies involved in computer vision is detailed.

2.1 Sports applications

GPS allows users to have feedback about their position, the distance they have covered, the time spent during a hike, speed information along their track and so on. It is the main feature that has made sports applications be so popular over the past few years, which has turned into an increase of the people going out and practicing long distance sports such as running, cycling or even just hiking.

One extended application of GPS in sports, are GPS watches. They are used among runners and they keep track of the distance, speed and time during the run. Once the training is done, the device stores all the data of the training. After that, the user can plug the watch in to the computer and be able to visualize the route in a map, the time spent for the whole route, per mile and much more feedback.

GPS is also one of the many features present in mobiles. This has lead into the emergence of tons of sports mobile application. The most common are running and cycling since these sports cover long distances, which is auspicious for the use of GPS.

Nowadays everyone has a smartphone and that is why these applications are so popular, even more than GPS watches. These applications have pretty much the same features than GPS watches but they give instant feedback once the training is done. The user is able to visualize the route within the mobile built-in maps right after it is finished or during the way.

2.2 Image registration

Image registration is the process of establishing the spatial or temporal correspondence among data sources. This data may be multiple images acquired at different times and/or with different modality. It is used in computer vision, medical image, automatic target recognition and in the analysis of images and satellite data. The registration is necessary to be able to compare or integrate the data obtained from these different measurements [2] [3].

In image registration there are normally four steps. First, a sufficient number of control points are required in order to estimate an optimal geometric transformation between two images (feature detection). After that, feature matching is necessary to find corresponding points because many feature points may be extracted from one image but not from the other. Then a transformation function is used to model the geometric relationship between the two images and finally, the unknown parameters of the function can be computed according to the control points [4].

Among data images, one of them is known as the reference and another image is known as the target. Image registration is a process of finding the spatial transformation of the target image to align it with the reference image and there are two fundamental approaches to achieve this transformation. Intensity based methods compare the intensity patterns in the images through correlation metrics, whereas feature based methods find correspondence between image features, such as points, lines and contours. Intensity based methods register complete images or sub-images. If sub-images are registered, centers of the corresponding sub-images are considered as corresponding feature points. Feature-based methods establish the correspondence between several points in images. Knowing the correspondence between several points in the images, a transformation is determined to map the target image to the reference images, establishing point by point, the correspondence between the reference images and the target.

Image registration algorithms can also be classified according to the transformation models that they use to relate the target image space to the reference image space. Rigid transformations are the simplest category; it includes linear transformations, which are translation, rotation, scaling and other affine transformations. Linear transformations are global in nature; therefore, they cannot model local geometric differences between images.

Apart from rigid transformations, there is another category that allows elastic or non-rigid transformations. These transformations are able to locally deform the target image to align it with the reference image. Non-rigid transformations include radial basis functions, continuous physical models and large deformation models.

An example of transformation, which is in fact important in this project, is the distance transform. A distance transform is a derived representation of a digital image, normally only applied to binary images. Each pixel of the image is labeled with the distance to the nearest obstacle pixel. A most common type of obstacle pixel is a boundary pixel, which

means a different value pixel, in a binary image. The result of the transform is a gray level image that looks similar to the original one, except that the gray level intensities of points inside foreground regions are changed to show the distance to the closest boundary from each point [5] [6].

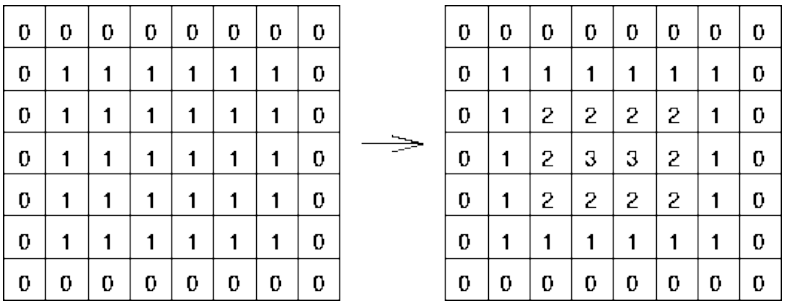
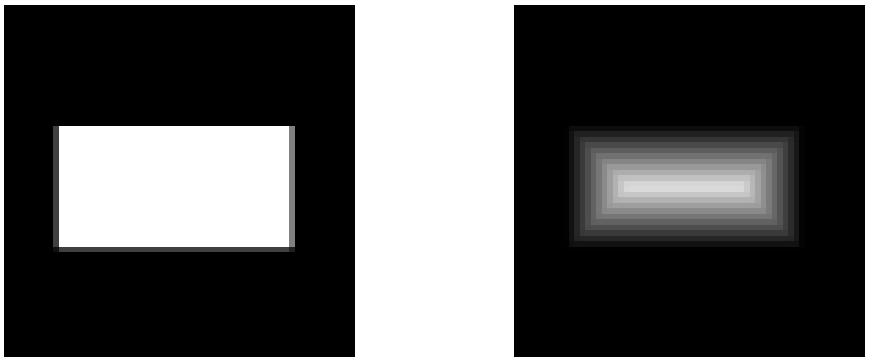


Figure 2.1: Example of how pixel values are assigned in a distance transform [5]

.

A representation of the example above is the following:



(a) Original image. (b) Distance transform image.

Figure 2.2: Example of the distance transform applied on a binary image [6]

.

While the predominant number of approaches is performed in the spatial domain, there are a few approaches that transform the data into frequency domain to solve a particular registration task. Spatial methods operate in the image domain, matching intensity patterns or images characteristics. Some of the algorithms that match characteristics are derivations of the traditional technics of manually registering an image, in which an operator decides the corresponding control points in the images. When the number of control points exceeds the minimum required to define an appropriate transformation model, the iterative algorithms such as RANSAC (Random Sample Consensus) can be used to robustly estimate the parameters of a particular type of transformation in image registration. Methods in frequency domain find the transformation parameters for the image registration while working in the transformation domain. These methods

work by simple transformation, such as translation, rotation and scaling. For example, applying the phase correlation method to a pair of images produces a third image, which has peak intensities at locations where the two images match the best. The intuition behind this method is that the resulting image will have the maximum peak value when the contents of the images math up exactly and that peak location correspond to the relative translation between the images [7]. Unlike many algorithms in the spatial domain, phase correlation method is noise, occlusions and other defects typical of medical or satellite images, insensitive. Besides, the phase correlation uses the Fast Fourier Transform to calculate the cross-correlation between the two images, which results in great performance improvements. This method can be extended to determine the rotation and scale differences between two images.

Another classification can be made between monomodal and multimodal methods. Monomodal methods register images in the same modality acquired by the only type of scanner/sensor, while multimodal methods tend to register images obtained by different types of scanners/sensors. Multimodal registration methods are often used in medical imaging since the images of a patient are obtained by frequency from different scanners.

Registration methods can be classified according to the level of automation they offer. There have been developed manual, interactive, semiautomatic and automatic methods. Manual methods provide tools to manually align images. Interactive methods reduce user bias by performing certain key operations automatically while still relying on the user to guide the registration. Semiautomatic methods make more registration steps automatically but they depend on the user to verify the correctness of a registration. Automatic methods do not allow user interaction and make all registration steps automatically.

A measure of image similarity quantifies the degree of similarity between the intensity patterns of two images. The choice of a measure of image similarity depends on the modality of the images to be registered. The most common examples of image similarity measures include cross-correlation, mutual information, sum of squares of intensity differences, and image uniformity ratio. Cross-correlation, sum of the squares of intensity differences and image uniformity ratio are used for registering images in the same modality. On the other hand, mutual information and normalized mutual information are the most popular image similarity measures for multimodal images registration. An example of similarity measures application is shape matching. Shape matching is highly related to similarity measures because it deals with transforming a shape, and measuring the resemblance with another one using some similarity measure. So, shape similarity measures are an essential ingredient in shape matching [8].

2.2.1 Applications

Image registration has applications in remote sensing (cartography update) and computer vision. Because of the many applications to which image registration can be applied, it is impossible to develop a general method that is optimized for all uses.

Registration of the medical image (for data from the same patient taken at different times, such as the detection of changes or control of a tumor), often also involves elastic registration to make face deformation of the patient (due to breathing, anatomical changes and so on). The non-rigid medical imaging record can also be used to record a patient's data for an anatomical atlas. It is also used in astrophotography to align images taken from space. Using the control points, the computer performs transformations on an image so that the main features are aligned with a second image.

Image registration is an essential part of creating panoramic images. There are many different techniques that can be implemented in real time and run on integrated devices such as cameras and camera phones.

CHAPTER 3

Method

The developed system finds in a map of pedestrian-accessible areas a path that makes a desired visual pattern depicted as a simple line drawing.

The algorithm consists of two major stages: one is the image comparison made through registration and the second stage is the remapping of the figure desired to represent onto the map.

3.1 Image comparison

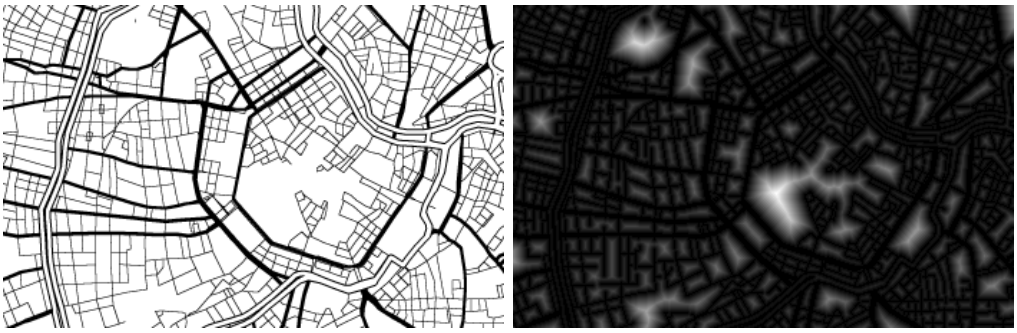
In the first part of the algorithm, the comparison between both, the map and the figure image is made. The outcome of this stage is to generate a new gray scale image in which each pixel is labeled with a different level of intensity. The lowest intensity value areas of the new image (that means the darkest parts) encode a higher probability to find a representation of the figure in that area. This stage can be understood as a similarity evaluation.

The comparison is applied on two bitmap images: one is the figure intended to be found in the map represented as simple line drawing as shown in Figure 3.1a and the other one is the map. This map image is obtained from Google Maps but getting an image from the default map that appears in the Google Maps application to use in the comparison would be useless. That map has many different colors, shapes, names of streets, roads and places that would introduce many errors during the similarity evaluation as we might be comparing pixels of the figure with pixels of the name of a certain street. To solve that, a new map in which only the street layer remains visible and in which there are not different shades or colors must be generated and it is done via the Google Maps Application Programming Interface (API). Using the Google Maps API by following the steps described in Chapter 4, we are able to generate a map with the described features, shown in Figure 3.1b.

(a) Sample figure that could be used in the comparison. (b) Generated map using Google Maps API.

Figure 3.1: Example of a sample image and map that could be used in the algorithm.

The comparison is not directly made over two images like the ones above, comparing two bitmaps would be very sensitive. Unless the figure perfectly matches some area in the pattern of the street layer, the differences would be very high. Even if the alignment between the figure and the street layer is nearly perfect, at the moment in which a pixel of the figure and a pixel of the map that does not correspond to any street are compared, it will result in a high difference as well. To solve this undesired high sensitivity of the similarity transform generated in this stage, instead of a bitmap image of the map we use its distance transform image like the one in Figure 3.2b. This way, during the similarity evaluation, even if the alignment between the figure and the street layer is nearly perfect, it will result in a high similarity score due to the much lower difference between the images being compared, which is the desired behavior for a nearly perfect aligned figure. The distance transform allows to better obtaining a measure of how close the figure being compared is to the each street in the map grid, as each pixel in the new map indicates the distance to its nearest street.



(a) Gray scaled map. (b) Map's distance transform.

Figure 3.2: Example of the distance transform applied on a gray scaled map previously generated.

After the similarity evaluation, a new image bitmap is created. As already explained,

each pixel of this new image indicates the similarity level of the figure at the position indicated by that pixel. A low pixel value, which corresponds to the dark parts, means a high similarity score. This image is obtained as a result of the comparison of the figure image all over the map distance transform.

During the whole process, the figure image is centered on each pixel of the map distance transform so that the figure image fits in the map and the comparison is carried out. That means that the figure image is not centered on pixels in which part of the figure would remain out of the map image bounds because there would be a part that could not be compared. It makes no sense to compare part of the figure because the aim of this comparison is finding a representation of the whole figure. During each comparison, the computed value is assigned to a new image that will correspond to the result similarity image. If the figure image is centered over the map distance transform and iterates all over it row after row starting on the top left corner, the new image should be filled following the same directions, row after row in this case. In the end, the similarity image dimensions would be the difference between the figure and map image dimensions.

The similarity image pixel values are obtained by comparing both the figure image and the corresponding part of the map distance transform. In this comparison the positions for black pixels in the figure are taken and used to obtain the values of the pixels corresponding to the figure in the distance transform. For each of the figure pixels we now have a value that indicates how far they are to their closest street in the map image and by computing the sum of those values we have an indicator of how close the figure is to its representation on the map streets. The higher this value is, the worse the similarity as it indicates that the figure is far from being represented in the street layer.

After assigning every value to the new bitmap, we may have an image like the one in Figure 3.3. The darker parts indicate high similarity score as already said. If the image generated is too uniform that nothing can be distinguished, we can vary every pixel intensity until much more differentiated areas appear.

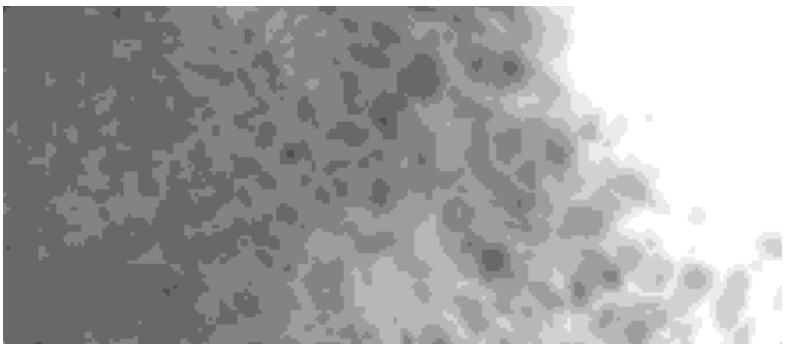


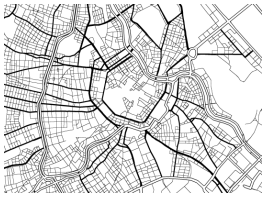
Figure 3.3: New gray scale similarity image obtained from the comparison between the map distance transform and the figure.

The comparison made so far only uses the figure image with the default orientation and

scale, but the figure may be found in the map with a different angle of rotation and size. In order to support that, it is necessary to add it into the comparison. In each iteration, the figure image is rotated and scaled so that the sum of the distance values is computed for every orientation and scale combination. At the end of the iteration the lowest value, which is the best result as previously described in this chapter, is used for the new image generated and stored with the corresponding orientation and scale.

As well as the figure image may have different scales, so does the map. Nowadays, digital maps have different levels of scale; the more we zoom in on a map, the more streets appear on the screen and the more we zoom out on a map, the fewer streets we are able to see. This is appreciated in Figure 3.4 and means that the whole comparison described so far needs to be done for different map scales in order to cover a wider range. In fact, this improves the chances of finding the figure in the map.

(a) Zoom level 1.



(b) Zoom level 2.

(c) Zoom level 3.

Figure 3.4: Different zoom levels for a map.

3.2 Remapping

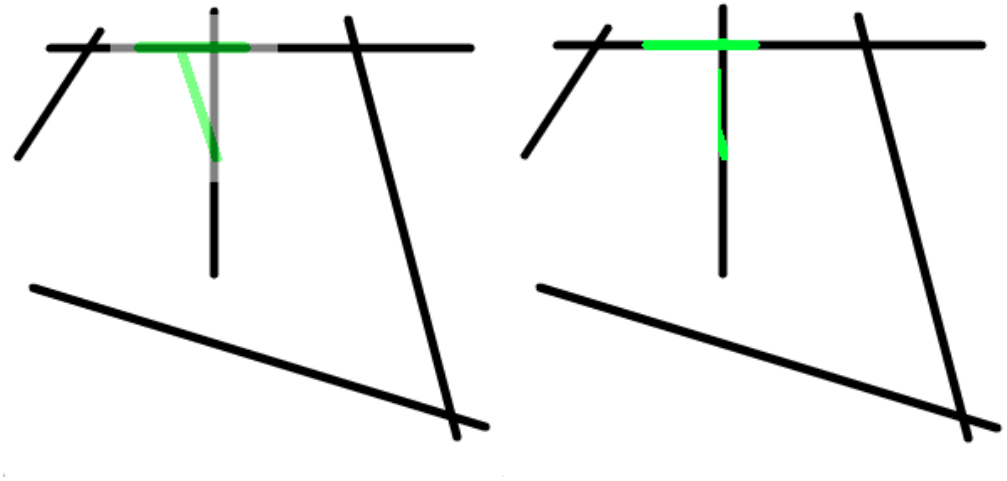
The comparison between the images generates a new gray scale map indicating the parts where the chances of finding a representation of the figure are higher. After that, the remapping should be done; this means moving the figure onto the streets in the map grid and this is achieved using the distance transform map.

For every position in the new gray scale image, besides the value there is information about the figure orientation and scale. This information is enough to do the remap. For a given position in the map, the corresponding figure image with its orientation and scale in that position is computed in order to obtain the positions for the figure black pixels. Then, the map distance transform image is used to do the remap.

In the distance transform image, a pixel having a value of 12 means that this pixel is 12 positions (pixels) away from the nearest pixel that has value 0, which corresponds to a

street. Only with the distance transform there is no clue on which direction to follow to get to the pixel with value 0. As the distance indicated by the distance transform is the distance to the nearest zero value pixel, the direction to follow will be the one with greatest rate of variation and that is given by the gradient. So in order to find the direction to follow, the gradient in that pixel must be calculated. Once having the direction, the zero value pixel can be reached going as many positions as the number indicated by the distance transform pixel value in that way.

Following that process for every of the black pixels positions will lead to a bunch of new positions in the map which will correspond to the new remapped figure as shown in Figure 3.5b.



(a) Map and figure overlayed. (b) Figure remapped.

Figure 3.5: Example of the a figure remapping into a map for a given position.

CHAPTER 4

Implementation

The system developed has two main parts, one is the development of the algorithm and the second is its implementation within an android app. In order for the algorithm to work properly, a previous set of maps must be generated also. In this chapter the implementation of every part is described.

4.1 Google Maps

As it is explained in Chapter 3, the maps used during the algorithm cannot be taken directly from a default map in Google Maps, they need of a previous processing so we are able to compute their distance transform and use it in the comparison with the figure image.

This issue has been solved in a very easy way using the Google Maps API. The Google Maps API enables creating our own custom maps just with a few lines of code. It consists on a simple html file that executes two scripts. An example of how the html file looks like is available at the Google Developers Website [9]. It is possible to generate the code for the map styles in a very interactive way at Google Developers Website [10]() and then paste it to the html file or directly write the styles in the html file.

In the example provided, there is a YOUR API KEY tag that must be replaced with an API key. It is possible to create one at Google API Console [11] and paste it in its corresponding place. The steps to follow in order to create the API key are described at the Google Developers Website [12].

4.2 OpenCV

4.2.1 Overview

The algorithm developed is extremely related to image registration as already explained. There are a lot of transformations such as scaling, rotating or the distance transform applied on the maps. Image processing has an important role and one of the technologies that provide all of this is Open Source Computer Vision (OpenCV). OpenCV provides everything necessary related to computer vision and has been the technology chosen for this project.

OpenCV is a library of programming functions mainly aimed at real-time computer vision as it is the current case. Originally developed by Intel research center, the library is cross-platform and uses the open sources Berkeley Software Distribution (BSD) license. OpenCV is free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac Operating System (OS), iOS and Android. It was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing.

For this project, C++ has been chosen as the development language. The algorithm has been developed using that programming language and later integrated into an Android mobile app, which uses Java as development language.

Regarding the algorithm implementation, all the functions related to image transformations has been developed using OpenCV built-in functions. These functions include scaling, rotating and computing the distance transform of an image. There have been necessary some minor enhancements in order for the functions to work as expected.

Images in OpenCV are treated as multichannel matrices. As it has been already said, the algorithm uses gray scale images, so these matrices will have one channel only; RGB is not necessary.

4.2.2 Installation

This project has been developed on Mac OS, so here is a description of the steps to follow in order to install OpenCV libraries and be able to use them [13].

Install Xcode

Download and install Xcodes from the Mac App Store.

Install command line tools

Start Terminal and copy and paste the following command:

```
xcode-select --install
```

Install hombrew

Install a package manager such as homebrew which is a matter of copying and pasting a single line from the Homebrew Website [14] into Terminal.

Find packages

Look for packages called "opencv":

```
brew search opencv
```

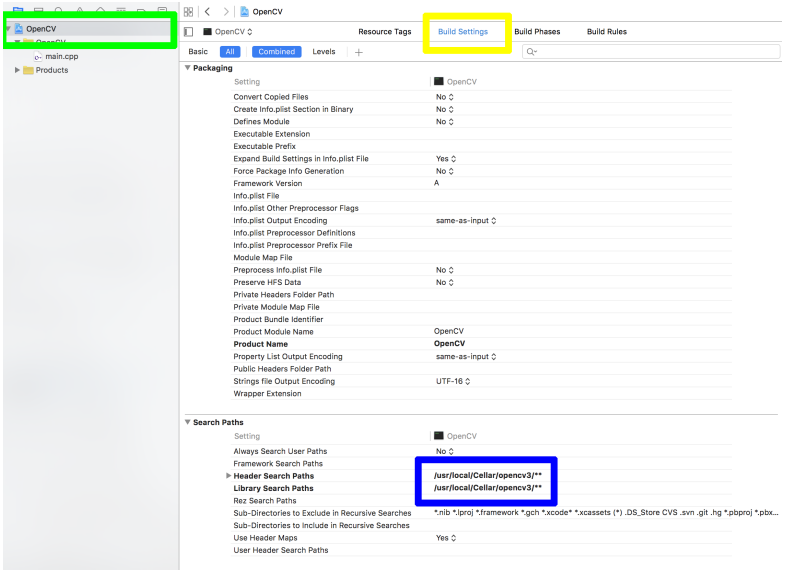
Install OpenCV

Install opencv package:

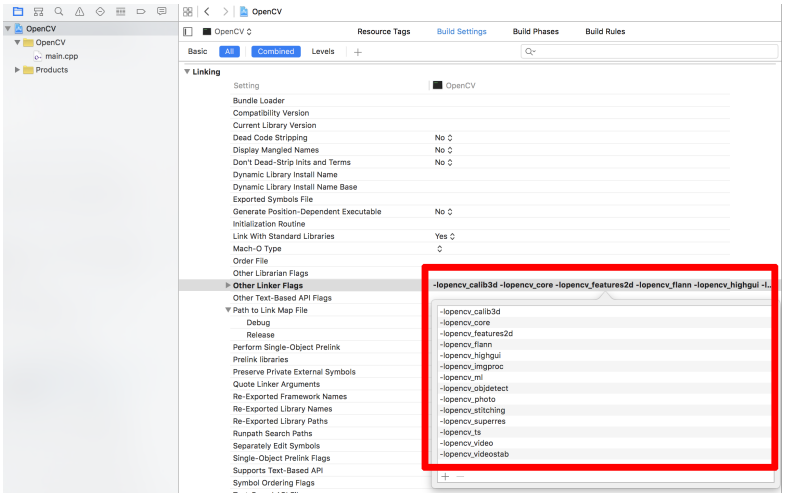
```
brew install opencv
```

Build with Xcode

It must be set in Xcodes where the header files are and also where the libraries are and which libraries to link. This will vary with the OpenCV version, but the places marked in the two diagrams below must be altered. It is easy clicking them in order - green area first, then the yellow, then the blue and the red.



(a) Set OpenCV paths.



(b) Set OpenCV linke flags.

Figure 4.1: Steps to follow in order to use OpenCV in an Xcode project [13]

.

The actual information that will need to go in the Xcode settings areas marked above can be found by running the pkg-config command.

To get the location of the header (include) files:

```
pkg-config --cflags opencv
```

To get the information needed to fill in for the linker in Xcode:

```
pkg-config --libs opencv
```

4.3 Android

4.3.1 Overview

In the second part of the project, the algorithm developed is included in a mobile application so it is accessible to users. The operating system chosen has been Android because of its wide range of users.

Android is a mobile operating system developed by Google, designed primarily for touchscreen mobile devices. The programming language used to develop Android apps is Java, so it has been the chosen one in this part of the project. It also enables to run native C/C++ code, which has been useful because of the first part of the project has been written in C++.

In order to include the algorithm code into the app, OpenCV offers Android and iOS libraries, so every OpenCV functionality is available in Java for Android and Objective-C for iOS, although C++ code can also be run.

The first implementation of the algorithm within the app was not made using native C++ code; instead, Android libraries provided by OpenCV were used. This has two inconvenients; first, the algorithm code has to be completely rewritten and second, it is much slower than running native C++ code.

Finally, it was decided to use native C++ code, which was a little harder to implement along with the Java code. In the next section the steps to follow in order to use OpenCV and native code are explained

4.3.2 Installation

Install Android Studio

Android Studio can be downloaded from the Android Developer Website [15].

Add OpenCV to Android Studio project

A tutorial [16] available in Youtube explains everything necessary to use OpenCV in an Android app.

Run OpenCV with native code in Android app

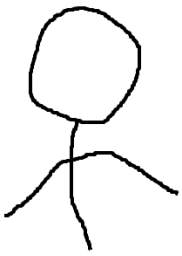
Another tutorial [17] available in Youtube explains the steps to follow in order to run OpenCV in native code within an android app.

CHAPTER 5

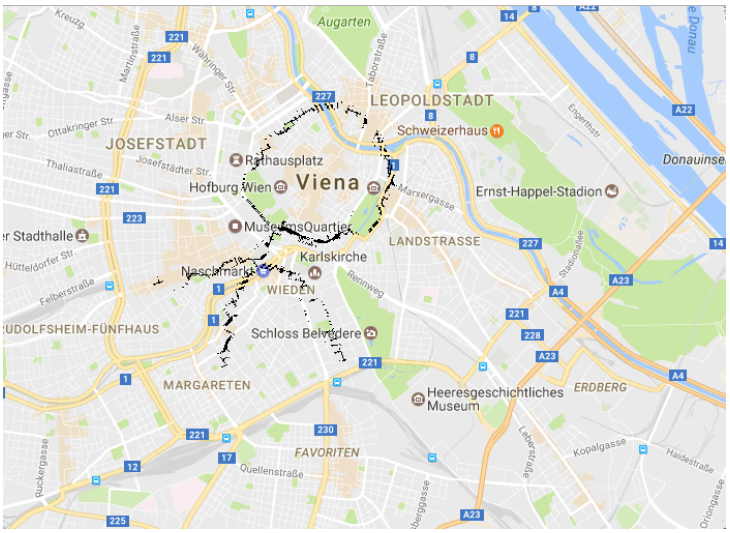
Results

In this chapter it is presented how the system developed represents a figure into a given map. First, there is a representation of a stick figure into the map of Vienna just to see how an image a bit complex is represented and secondly, it is shown how a figure is remapped from what it would like without remapping.

In the following images we see the representation of the drawing in Figure 5.1a in the map of Vienna. The result corresponds to Figure 5.1b. The drawing can clearly be seen in the representation but there are some parts of the remap that go across zones that are not walk able. That is something to comment in the last chapter.



(a) Sample figure to represent.



(b) Representation of the sample image in a map.

Figure 5.1: Example of a sample image represented in a map.

In the next following images we are able to see how the remap works. In Figure 5.2a a figure is placed over a given map with the corresponding size and scale in that position after having computed the comparison algorithm. Then it should be remapped onto the closest streets and the result can be seen in Figure 5.2b.

(a) Sample figure overlaid on top of a given map with the corresponding size and scale.

(b) Remap of the overlaid figure.

Figure 5.2: Example of how the remap of a figure works at a particular position.

CHAPTER 6

Conclusion and outlook

6.1 Summary and critical review

This thesis began with a single idea: Try to represent a drawn figure in the streets map layer of a city in an automated way. After doing a preliminary research, nothing similar was found. Everything done until then was figures represented by artists that had come up with the path in a certain region.

A first approach was carried out by comparing, both the image of the corresponding map and the figure. This comparison was made over the distance transform of the images. The results obtained were not that good as using the distance transform in both images, the number of dark pixels in the figure increase and the sums of pixel values differences along the map were pretty uniform.

The second approach improved the results. Instead of using the distance transform of the map and the image, only the map one was used and only the black pixels in the figure were taken to compute the image with the sum of pixel differences. This made this image to have much differentiated parts, which is better.

Regarding the implementation part, the first technology chosen to develop the algorithm was MATLAB but the results were not good at all. In terms of time consuming, it was really slow so it was decided to change to C++ and use OpenCV libraries. Time consuming improved a lot, although it is still slow due to the amount of calculations that are being made between the images.

As a final reflection on this thesis, the main conclusion would be that the method performed is good as a first approach. The results demonstrate that the figure shape is more or less remapped onto the streets layer. This work has served to confirm that the initial idea was achievable and opens a new path in which it can be improved.

6.2 Future work

The first approach of the system presented in this thesis opens a new path for improving it and adding new features. The algorithm developed focuses on the basics, which is translating a figure into the streets map layer without any conditions.

The system treats every part of the map as equal, that is to say; for example, in a map a building, a square or a park may look similar (as an empty space) but this fact is not relevant for the algorithm. A next direction to follow could be somehow to penalize positions of the map weather they are walk able or not, so the remapping does not send the user over a building for example.

In terms of performance, there is another way in which the system could be improved. Currently it takes a long time to do all the calculations, especially if the images used are big, so it would be a good idea to optimize it and lower the time of execution. A good direction to follow may be to use gpu computing.

Glossary

GPS art or GPS drawing is a method of drawing that uses GPS technology to create large-scale artwork. It combines art, movement, and technology. 1, 5

open source Decentralized development model that encourages open collaboration. Its main principle is peer production, with products such as source code, blueprints, and documentation freely available to the public. 18

Xcode Integrated development environment for macOS containing a suite of software development tools developed by Apple for developing software for macOS, iOS, watchOS and tvOS. 18, 19

Acronyms

API Application Programming Interface. 11

BSD Berkeley Software Distribution. 18

GPS Global Positioning System. 1, 5

OpenCV Open Source Computer Vision. 18, 19, 21, 23

OS Operating System. 18

Bibliography

[1] A. Gri. Artist draws world's largest doodles by riding his bike with gps. [Online]. Available: http://www.boredpanda.com/bike-gps-doodle-stephen-lund/

[2] (2017, April). [Online]. Available: https://en.wikipedia.org/wiki/Image_registration

[3] J. F. Barbara Zitová, "Image registration methods: a survey," June 2003.

[4] Z. X. . Y. Zhang, "A critical review of image registration methods," *International Journal of Image and Data Fusion*, 2010.

[5] (2016, March). [Online]. Available: https://en.wikipedia.org/wiki/Distance_transform

[6] (2003). [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/HIPR2/distance.htm

[7] [Online]. Available: http://www.cs.utah.edu/~ssingla/IP/P4/Index.html

[8] R. C. Veltkamp, "Shape matching: Similarity measures and algorithms," Dept. Computing Science, Utrecht University, Tech. Rep.

[9] [Online]. Available: https://developers.google.com/maps/documentation/javascript/styling?hl=es-419

[10] [Online]. Available: https://mapstyle.withgoogle.com

[11] [Online]. Available: https://console.developers.google.com

[12] [Online]. Available: https://developers.google.com/maps/documentation/javascript/get-api-key

[13] (2015, December). [Online]. Available: http://stackoverflow.com/questions/34340578/installing-c-libraries-on-os-x

[14] [Online]. Available: https://brew.sh

[15] [Online]. Available: https://developer.android.com/studio/index.html?hl=es-419

[16] [Online]. Available: https://www.youtube.com/watch?v=nv4MEliij14&list=PL6v5F68v1ZZzTDq8VI9Jcmb0J99WRrYn4&index=6

[17] [Online]. Available: https://www.youtube.com/watch?v=Oq3oiCfSgbo&list=PL6v5F68v1ZZzTDq8VI9Jcmb0J99WRrYn4&index=4