JACOBS
UNIVERSITY

School of Engineering and Science

Bachelor's Thesis

Automatic marker-free registration of terrestrial laser scans based on SIFT features from reflectance images

Darko Makreshanski

May 2010

Supervisor: Prof. Dr. Andreas Nüchter

Abstract

Registration in computer vision terminology is a process in which multiple data sets are joined together under one reference system. The question of registration of 3D point clouds from terrestrial laser scans is still open in the scientific world and many approaches have been offered, each with its own advantages and drawbacks. This guided research aims to investigate into a process of registration of terrestrial laser scans based on scale invariant features extracted from a panorama image generated from reflectance values. This process is fully automatic, does not rely on any additional information on the estimate of the poses such as odometry or GPS, does not involve obtrusion of the scene by placement of special markers and should offer registration of scans that are relatively well separated from each other. This guided research deals with developing and benchmarking this method for efficiency and robustness

Acknowledgements

I would to express my deepest gratitude to my supervisor Prof. Dr. Andreas Nüchter for his invaluable constant support and guidance during the entire guided research. I would also like to thank Dorit Borrmann and Jan Elseberg for their support and for providing the Bremen downtown dataset.

Contents

1 Introduction

As the technology is advancing, laser range scanners become more accessible and thus 3D point clouds manipulation from terrestrial laser scans is an interesting field for research. This guided research focuses on the registration of such terrestrial scans based on image features of a panorama generated from the scans. The current general approach to registration of point clouds is consisted of two steps: a rough step and a fine step. The rough step can be done either manually by hand, by some sort of help through external sensors like odometry, GPS, etc., or fully automatically by an algorithm that works only on the scan data. The fine alignment step consists of fine adjustment and minimization of inter-point distance, which is usually done with help of the well known ICP algorithm [BM92].

The process of registration of this research attempts to tackle the rough alignment step and provide a fully automatic registration without the help of external sensors, and without the need of placing markers on the scene. The idea of the research is to provide a method where a person could scan an area from several viewpoints with a terrestrial laser scanner and afterwards input these scans to the system and obtain a full 3D point cloud model of the scanned area. This project is based on the Riegl VZ-400 scanner [Rie09]. This scanner, like most terrestrial laser scanners, apart from measuring the time-of-flight for each emitted ray, also measures the reflectance value, i.e. the amount of energy that returns from the ray. These reflectance values are used to generate a panorama image for each scan, which provide a robust ground for detecting image features. For this purpose the popular SIFT method [Low99, Low04] for extraction of robust scale invariant image features is used with the help of an existing open source implementation [Now05]. The matching process also takes into consideration the depth at each feature point, obtained by the time-of-flight information for each point, which is used with a RANSAC filtering algorithm [FB81] to eliminate the outliers. The filtering algorithm is essential in this case, since the number of extracted features, that is around 10,000 for a 1440x400 image, is large and the number of outliers is also relatively large.

We have also provided a sound benchmark of the robustness of the SIFT features for registration of terrestrial scans, and we have tested the system for performance and efficiency with respect to the optional parameters it offers.

2

State of the Art

Iterative algorithms

Probably the most popular and most cited algorithm of point cloud registration is the ICP (iterative closest point) algorithm [BM92] which works in the way that given an initial value of the relative distance of the scans it creates pairs of points between the scans and computes an estimate of the transformation, and this procedure is repeated until convergence is reached. As the name suggests, the pairs of points are found by getting the closest point of the other dataset. Obviously the success of this algorithm directly depends on the accurate identification of point pairs, and since they are found by means of smallest distance, the initial distance between the scans is required to be relatively small, i.e. an initial rough information about the pose estimate of the scans is required. A lot of modifications and extensions to this algorithm exist. ICCP (iterative closest compatible point) algorithms [GRB94] [GA05] developed a least squares method for registration of 3D scans. There exist some modifications to this algorithm that increase the radius of convergence, but the basic problem of the algorithm still remains.

Nevertheless, ICP is frequently combined with other algorithms to provide a final fine alignment of the scans.

Additional sensors

Other methods providing a rough alignment of the scans are based on additional sensors. These types of systems usually utilize a georeferencing system composed of GPS, compass and pan/tilt measuring devices [TAB$^+$04,SB05]. These systems obtain good results when combined with ICP for fine registration, however their drawback is the dependence of quality and availability of the external sensors.

EGI

Another set of algorithms [Dol05,MPD06] are based on EGI (extended gaussian images) [Hor84]. This type of algorithms are usually suited for the determination of the relative rotation between scans, but can also be extended for determining the translation as well. The rotation is determined by creating a orientation histogram of the scans and by searching through the rotation space to find the rotation with the largest correlation. Translation is estimated in a similar way where the a maximum correlation of the scans is sought in the translation space. Obviously, this approach is very computation intensive and general techniques are to reduce the complexity by going into frequency-space. The problem of this approach is that it only works on point clouds, and thus is mostly suited for registration of single 3D objects, and is generally unsuitable for registration of urban landscapes where there are a lot of repetitive patterns.

Marker-based

The current standard for registration, available also in many commercial systems, is a simple method which involves placing of markers in the scanned scene. A minimum of three correspondent markers must be present in a pair of scans so that registration of the scans can occur. This is a very robust method as markers are usually highly identifiable and distinguishable, however the obvious drawback is the time consumptive placement of markers and also the intrusiveness to the environment.

Feature-based

In the recent years various approaches based on extracting features have been proposed. [hBL04] propose a registration method based on variation of curvature in the neighborhood as method for matching points. This method is suitable for close range indoor and landscape scans, however unsuitable for urban scenes where parallel and orthogonal planes are dominating. Another approach [DB06] is to process the point cloud and group points into patches of planes and match sets of planes across scans. The drawback is that it contains the expensive preprocessing step of determining of planes. Another and very popular way is to connect point clouds to 2D images taken from external camera and afterwards extract and match features from separate images that correspond to certain point clouds [BDW+04, BF07]. Currently the most popular method for extracting of image features is the SIFT method proposed by Lowe [Low99, Low04]. This method provides the most robust scale invariant and partially rotation invariant. This was shown in a survey by Mikolajczyk and Schmid [MS05]. As most of the terrestrial scanners also return a reflectance value as well, it is obvious the method with extracting image features from reflectance images can be used as well with the advantage that the step of aligning camera data and scanner data is skipped. This method is not novel and is currently very trendy. Böhm and Becker [BB07] have published a paper in 2007 explaining this method and showed a small narrow angle scans example. Wang and Brenner [WB08] have extended the work of [BB07] and proposed an addition to the SIFT descriptor to contain geometry features which reduced the number of outliers.

Kang et al. [KLZ+09] not aware of the work of the previous two papers also published a paper with a similar technique and added a technique for global registration of the scans.

2.1 Motivation

The motivation behind this research is to reimplement and extend the work of [KLZ+09] and [BB07]. Both of them have only considered registration of relatively small inter-scan distance, and and offer no real evaluation of this method. [WB08] provided some tests with multiple scans of a scene but only provided the relative distance and orientation as metric. One goal of this research was to provide various tests to the usage of the SIFT method on registration of terrestrial laser scans. The SIFT method is advertised to efficiently match features that are rotated up to 60 degrees and we have tested at what relative angles does the system actually makes most use of. We have also tested the percentage of inliers in various types of scans.

Another motivation was to examine another approach to the part of generation and processing of the panoramic image data in the registration process. Namely a dynamic resolution of panoramic images is introduced for the sake of increasing the speed of execution of the registration process and testing the effects on the precision and efficiency. The discrepancy between the image and the point cloud that has been introduced was handled by including the range data into the image object, but without including in the SIFT calculations. In this way we have completely excluded the part of processing of 3D point cloud data which is many times in the orders of hundreds of megabytes.

3

Methodology

3.1 OpenGL framework

Since the registration process proposed in this project is composed of a significant number of substeps, and each substep deals with computation of large amounts of data debugging even small code errors is very hard. For this reason we have created a tool with which we can test and see the output of each stage of the development. This tool is a simple OpenGL framework which can be easily extended to support displaying of the various data types.

The framework supports navigation throughout the scene similar to a free-look navigation in first person shooter games using 3 axes for the movement of the camera in the 3D space and 2 axes for the rotation of the camera in spherical coordinates. Camera movement is done via the keyboard and camera rotation is done via the mouse. This creates a very user friendly navigation and exploration of the scene.

The system supports visualization of single scans in 2D and 3D, visualization of the panorama maps and pair of maps with matched features and visualization of multiple registered scans.

3.2 Panorama generation

As the Riegl VZ-400 scanner also outputs the points in spherical coordinates the generation of the panorama image is a fairly easy process. The most simple procedure is to map points by just a correspondence between the spherical angle coordinates α and β from the scan and the Cartesian coordinates x and y on the image.

However things are more complicated since the scanner does not provide this correspondence between the scan in spherical coordinates and the panorama image, which breaks the simple mapping to the panorama image. Another issue is that we wanted the analyze the efficiency of the registration based on several resolutions of the panorama images, and for for this reason we have provided a method for creating panorama images with dynamical resolution.

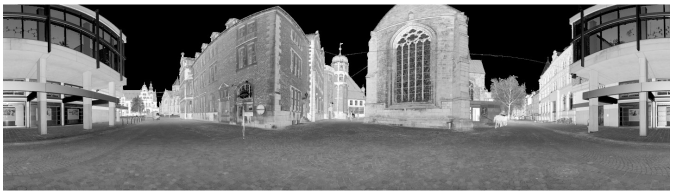
- First the panorama data object is initialized as two-dimensional map of $W \cdot H$ so-called super-pixels where W and H are the dynamic width and height of the panorama. All values for all super-pixels are initialized to zero.

- Super-pixels are pixels which apart from the regular value of the pixel, the reflectance in this case, they also contain a "meta" value which holds the range information that corresponds to the part of the scan that the pixel represents. The reason for this additional range data is to keep the pixel-to-point correspondence and to allow the system to completely eliminate the point cloud data from the process of registration.

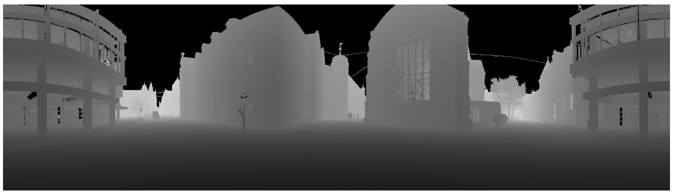
- After initialization the panorama object is populated with the following two-step process:

  1. First a parallel two-dimensional array with the same dimensions as the panorama is initialized with all values set to zero. Afterwards for each point from the scan the nearest corresponding super-pixel is found and then the reflectance and range values of the point are added to the corresponding fields of the super-pixel. Afterwards the corresponding value for the super-pixel in the parallel array is increased by one.
  2. In the second step, for each super-pixel the values for reflectance and range are divided by the amount of corresponding value in the parallel array.

This provides a rough and not so good method for resizing, in terms that it only works with reducing and not increasing the resolution. In fact, in our case we do not need to explore the case with increasing the resolution, as it will just decrease the speed and it will not provide any improvements because this problem of increasing the resolution will be solved by filling the values not only for the corresponding super-pixel but also its neighbors using Gaussian distribution as a weight function, and since the SIFT method is based on scale-space with the Gaussian distribution, it obviously will give the same results.



(a) reflectance map



(b) range map

Figure 3.1: An example pair of a reflectance and range maps

One issue that we had to tackle is the value range of the images. Namely the reflectance value has a higher dynamic range than a regular JPEG image, and most of the values recorded are only within a small region of this range, thus making the resulting images to be very low in contrast. To tackle this problem we have cropped areas of the range which had almost no values and extended the area with most details to fit the dynamic range of an image. The result is shown in below.

(a) before equalization

(b) after equalization

Figure 3.2: The effect of histogram equalization

3.3 SIFT features

SIFT or Scale Invariant Feature Transform is a method proposed by Lowe [Low99, Low04] which is currently the most popular and a very robust method for extracting of features from images. The SIFT features are invariant to scaling, translation and rotation and partially invariant to illumination changes and affine or 3D projection. For this project an open-source implementation for extraction and matching of SIFT features was used autopano-sift-C, a C port of the C# software autopano-sift [Now05]

3.3.1 Feature extraction

The SIFT feature extraction is consisted of several steps. First an image pyramid is formed of the image by filtering the image with the Gaussian function at multiple levels. This creates multiple images at various scales which are the basis of the scale invariance of the method. Afterwards the differences, so called Difference of Gaussians, of all the consecutive levels are computed and the local maxima of this function are potential keypoint positions. Afterwards, the potential keypoint positions are refined and the results are filtered by removing keypoints with low contrast and keypoints that correspond to edges.

The next step of the algorithm is to determine the orientation of the features. This is done by calculating a magnitude gradient $m(x, y)$ and orientation $\theta(x, y)$ of certain points which are precomputed for each image level L. Afterwards an orientation histogram of 36 bins is created of multiple orientations weighted by the magnitude gradients from points around the keypoint position. And finally the largest peak of the histogram is taken as orientation for the feature. If the histogram has multiple large peaks then these are treated as separate features with same position and different orientation.

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right)$$

After the determination of the position, scale and orientation of the keypoints, the area around the keypoint is described in a similar fashion as the orientation is determined. This method is based on a model of how biological vision works, in particular of complex neurons in primary visual cortex. Magnitude gradients and orientations are determined for a certain area around the keypoint position, which is by default of size 16x16 pixels. Coordinates of the gradients are rotated by the orientation value for the keypoint. Then these cells are weighted with a Gaussian function and finally grouped into 4x4 array of orientation histograms with 8 bins. This produces a feature descriptor for each keypoint of 4x4x8=128 dimensions.

3.3.2 Feature matching

Since the standard kd-tree search for high-dimensional spaces is not very efficient, a modified version of the kd-tree search called Best Bin First search [BL97] is used for matching SIFT features. This method stops after exploring a certain amount of closest bins in the kd-tree which results only in finding an *approximate* nearest neighbor, however tests have shown that in the majority of cases the nearest neighbor is returned.

In order to determine if a certain nearest neighbor is a positive match or not the SIFT method uses a method which relies on the ratio of the Euclidean distances between the keypoint and the nearest neighbor and the keypoint and the second nearest neighbor. It uses this method in the favor of the simpler method of a threshold on the distance with the explanation that some of the descriptors are more distinctive than the others, which means that it is difficult to choose a certain threshold value.

The default threshold for the ratio in the autopano-sift-C implementation was 0.6 which according to the test results in [Low04] eliminates most of the outliers, however it also eliminates a large percentage of inliers. We have modified this and for our purposes used a threshold ratio of 0.8 which should eliminate around 90% of outliers and 5% of inliers. We have increased this value because in our system outliers are not as scary as in the panorama generation software since we have one more step of filtration in the RANSAC process which also takes into consideration the range value for the keypoints.

3.4 Registration

3.4.1 2-scan registration

For registration of a pair of scans a RANSAC-like approach is used in this project. The algorithm takes as input a set of matches that is returned from the SIFT matching algorithm and the two panorama map objects that correspond to the maps which were used to extract the features. The algorithm goes through a subset of combinations of 3 point pair matches and works on the two triangles that are formed from these point pairs.

First it calculates the translation parameter between the triangles which is obtained by the distance between the centroids of the triangles. Afterwards it shifts the triangles so that the centroids are placed at the center of their reference frame and it calculates the rotation that minimizes the error between the points. The rotation is obtained with a closed form formula proposed by Horn [Hor87] which implementation was used from the SLAM

6D source code [NÖ9]. The formula is derived in [Hor87] and the final conclusion is that the rotation is represented in a form of a quaternion which is the eigenvector that correspondents to the maximum eigenvalue of the following matrix:

$$N = \begin{pmatrix} S_{xx} + S_{yy} + S_{zz} & S_{yz} - S_{zy} & S_{zx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & S_{xx} - S_{yy} - S_{zz} & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} - S_{xz} & S_{xy} + S_{yx} & -S_{xx} + S_{yy} - S_{zz} & S_{yz} + S_{zy} \\ S_{xy} - S_{yx} & S_{zx} + S_{xz} & S_{yz} + S_{zy} & -S_{xx} - S_{yy} + S_{zz} \end{pmatrix}$$

where:

$$S_{mn} = \sum_{i=1}^{3} p^a_{i_m} \cdot p^b_{i_n}$$

where a and b correspond to the first and second triangle respectively, and m and n to one of three coordinates x, y, or z. After the eigenvector $v = \begin{pmatrix} a & b & c & d \end{pmatrix}^T$ for the maximum eigenvalue of N is found, the rotation matrix R is defined as:

$$R = \begin{pmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2bd + 2ac \\ 2bc + 2ad & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2cd + 2ab & a^2 - b^2 - c^2 + d^2 \end{pmatrix}$$

And finally the affine transformation matrix for homogeneous coordinates that is used in this system is defined as:

$$M = \begin{pmatrix} & & & \\ & R & & v_t \\ & & & \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where v_t is the translation vector and is defined as:

$$v_t = R(-c_b) + c_a$$

where c_a and c_b are the centroids of the triangles a and b respectively.

After the transformation matrix M is obtained for the pair of triangles it is tested on how good it performs on the other point pair matches. If the error for a certain point pair is below a specified threshold t then this point pair is assumed to be an inlier. Afterwards if the number of inliers is above a certain threshold d, the transformation matrix is considered to be a valid transformation matrix and the finally the average of the errors of the inliers and the number of inliers is used to determine which transformation matrix provides the best transformation.

The pseudocode for this algorithm is presented below.

Algorithm 3.4.1: REGISTER2SCAN($matches, mapA, mapB$)

$best\_av\_error \leftarrow DOUBLE\_MAX$
$best\_count \leftarrow 0$
for each $p1, p2, p3 \in smatches \subseteq matches$
do {
  comment: use the range information in the maps to obtain 3D triangles
  $triangleA \leftarrow$ GETTRIANGLE($p1.A, p2.A, p3.A, mapA$)
  $triangleB \leftarrow$ GETTRIANGLE($p1.B, p2.B, p3.B, mapB$)

  $triangleA \leftarrow triangleA -$ CENTER($triangleA$)
  $triangleB \leftarrow triangleB -$ CENTER($triangleB$)
  $rotation \leftarrow$ GETROTATION($trianglA, triangleB$)
  $translation \leftarrow$ ROTATE($rotation, -centerB$) $+ centerA$

  $error \leftarrow 0$
  $count \leftarrow 0$
  for each $p \in matches \setminus \{p1, p2, p3\}$
  do {
    $coordA \leftarrow$ GETCOORD($p.A, mapA$)
    $coordB \leftarrow$ GETCOORD($p.B, mapB$)
    $transtcB \leftarrow$ ROTATE($rotation, coordB$) $+ translation$
    $terror =$ ABSOLUTEVALUE($transtcB - coordA$)
    if $terror < t$
      then {
        $error \leftarrow error + terror$
        $count \leftarrow count + 1$
  if $count > d$
    then {
      $av\_error \leftarrow error/count$
      if $av\_error - d\_inf \cdot count < best\_av\_error - d\_inf \cdot best\_count$
        then {
          $best\_av\_error \leftarrow av\_err$
          $best\_count \leftarrow count$
          $best\_translation \leftarrow translation$
          $best\_rotation \leftarrow rotation$
if not $best\_count = 0$
  then return (TRANSFORMATION($best\_rotation, best\_translation$))
  else return (**false**)

3.4.2 n-scan registration

Due to lack of time and only for demonstration purposes a simple technique for registration of multiple scans has been implemented. In the process, all the matched features are grouped into match-sets, where each match-set contains matches from one pair of images. This n-scan registration algorithm takes as input an object which contains a group of match-sets and it also takes as input a list of panorama map objects as described in 3.2. The algorithm goes through all the match-sets in the group and tries to register the scan pair with the 2-scan registration technique. If 2-scan registration succeeds it goes through the list of independent components and checks if the two scans are included in any component. This means that there are four cases to consider:

1. **both of the scans are not in any component**
   In this case the algorithm creates a new component and adds the first scan as a reference scan in this component, and the second scan along with the resulting transformation from the 2-scan registration step.

2. **both of the scans are present in some components**
   For this case there are two subcases: either they are in the same component, or they are in different components. If they are in the same component the algorithm does nothing, however if they are in different components the algorithm merges the second component into the first one. The merging technique is consisted with transforming the scan transformations from the second component in the frame of reference of the first component.

   Let A and B denote the first and second components respectively. Then let A_m and B_m denote the scans that were registered and tr_{AB} be the 4x4 transformation matrix that transforms the scan B_m into the reference frame of scan A_m. Then let trA_m be the transformation for scan A_m for the reference frame of component A, and trB_m the same for B_m and B. Then let B_i denote a scan in component B and its transformation trB_i that is supposed to be converted to the frame of reference of component A. And finally let x_{A_m}, x_{B_m} and x_{B_i} denote three points from the scans A_m, B_m and B_i respectively and which are in their scans' frame of reference. We assume that the three points correspond to one point in the real world and that all transformations are ideal. Thus, we have the following statements:

$$trAB \cdot x_{B_m} = x_{A_m} \tag{3.1}$$

$$trB_i \cdot x_{B_i} = trB_m \cdot x_{B_m} \tag{3.2}$$

   And we are looking for matrix M_i such that:

$$M_i \cdot x_{B_i} = trA_m \cdot x_{A_m} \tag{3.3}$$

So first we multiply (3.1) by trA_m and (3.2) by ${trB_m}^{-1}$ and we obtain:

$$trA_m \cdot trAB \cdot x_{B_m} = trA_m \cdot x_{A_m} \tag{3.4}$$

$${trB_m}^{-1} \cdot trB_i \cdot x_{B_i} = x_{B_m} \tag{3.5}$$

And then substituting x_{B_m} in (3.4) we obtain:

$$trA_m \cdot trAB \cdot {trB_m}^{-1} \cdot trB_i \cdot x_{B_i} = trA_m \cdot x_{A_m} \tag{3.6}$$

And finaly by comparing (3.3) and (3.6) we conclude that:

$$M_i = trA_m \cdot trAB \cdot {trB_m}^{-1} \cdot trB_i \tag{3.7}$$

So the merging step is to move all the scans B_i from component B to component A with a transformation matrix M_i

3. **the first scan is found in some component**
   In this case the second scan needs to be added the this component A and the matrix which will transform it to the frame of reference of A will be:

   $$M = trA_m \cdot trAB \tag{3.8}$$

   where trA_m is the transformation matrix for the first scan which transforms it for the frame of reference of A, and $trAB$ is the result of the registration of the two scans which transforms the second scan into the frame of reference of the first scan

4. **the second scan is found in some component**
   This case is similar to the previous case with the exception that $trAB$ needs to be inverted so that it transforms the first scan into the frame of reference of the second scan.

   $$M = trB_m \cdot trAB^{-1} \tag{3.9}$$

This provides a simple global registration technique in which errors sum up because it does not have any optimizations involved. It serves primarily for demonstration purposes. The pseudocode of the algorithm is presented below.

Algorithm 3.4.2: REGISTERNSCAN(*matchesGroup*, *maps*)

```
components ← TRANSFORM[][]()
for each matches ∈ matchesGroup
  do
    result_tr ← REGISTER2SCAN(matches, maps[matches.A], maps[matches.B])
    if result_tr = false
      then CONTINUE()
    for each component ∈ components
      do
        for each tr ∈ component
          do
            if tr.id = matches.A.id
              then
                componentA ← component
                trA ← tr; foundA ← true
            if tr.id = matches.B.id
              then
                componentB ← component
                trB ← tr; foundB ← true
    if not foundA and not foundB
      then
        newtrA ← TRANSFORM(); newtrA.id ← matches.A.id
        newtrB ← result_tr; newtrB.id ← matches.B.id
        newcomponent.ADD(newtrA)
        newcomponent.ADD(newtrB)
        components.ADD(newcomponent)
    else if foundA and foundB
      then
        if not componentA = componentB
          then
            for each tr ∈ componentB
              do
                newtr ← trA * result_tr * INVERSE(trB) * tr
                newtr.id ← tr.id
                componentA.ADD(newtr)
            components.REMOVE(componentB)
    else if foundA
      do
        newtr ← trA * result_tr; newtr.id ← matches.B.id
        componentA.ADD(newtr)
    else if foundB
      do
        newtr ← trB * INVERSE(result_tr); newtr.id ← matches.A.id
        componentB.ADD(newtr)
return (components)
```

4

Experiments and results

Testing environment

All experiments and tests were done on a machine with Intel®Core^{TM}2 Duo CPU T8300 @ 2.40GHz and 2GB RAM.

4.1 Bremen downtown dataset

Testing dataset

This dataset is consisted from 13 scans made in downtown Bremen, Germany. The reflectance maps of the scans are shown in figure 4.1 and the scanner's positions are shown in figure 4.2. The scans were made with angular resolution of 0.04°, which gives a theoretical resolution of $9000x2500$.

4.1.1 Pairwise registration results

The following section offers a set of tests that evaluate the robustness of the SIFT features for the purpose of registration of point clouds and other tests for efficiency, performance and precision of the method. Evaluation of SIFT feature extraction performance is available in table 4.1. Evaluation of feature matching and registration performance along with the ratio of inliers and outliers is available in table 4.2. The number of inliers was determined by the location of the highest peak in the histogram for the amount of inliers for all combinations of point pair triangles. This histogram was obtained with a pairwise registration with the t parameter set to 1.0 for $1440x400$ and $2160x600$, and 0.5 for $2880x800$ and $3600x1000$ resolutions. The value of the t parameter was decided with the help of the histogram 4.3 which shows the frequency of the errors of all the matches with respect to the best registration and compared with different resolutions. Another histogram 4.4 shows the robustness of the SIFT features in this scenario. This histogram shows the frequencies of the relative angles for the inlier matches that are formed by the position of the feature and the positions of the scaner. And finally the resulting pairwise transformations for consecutive scans are shown in table 4.3

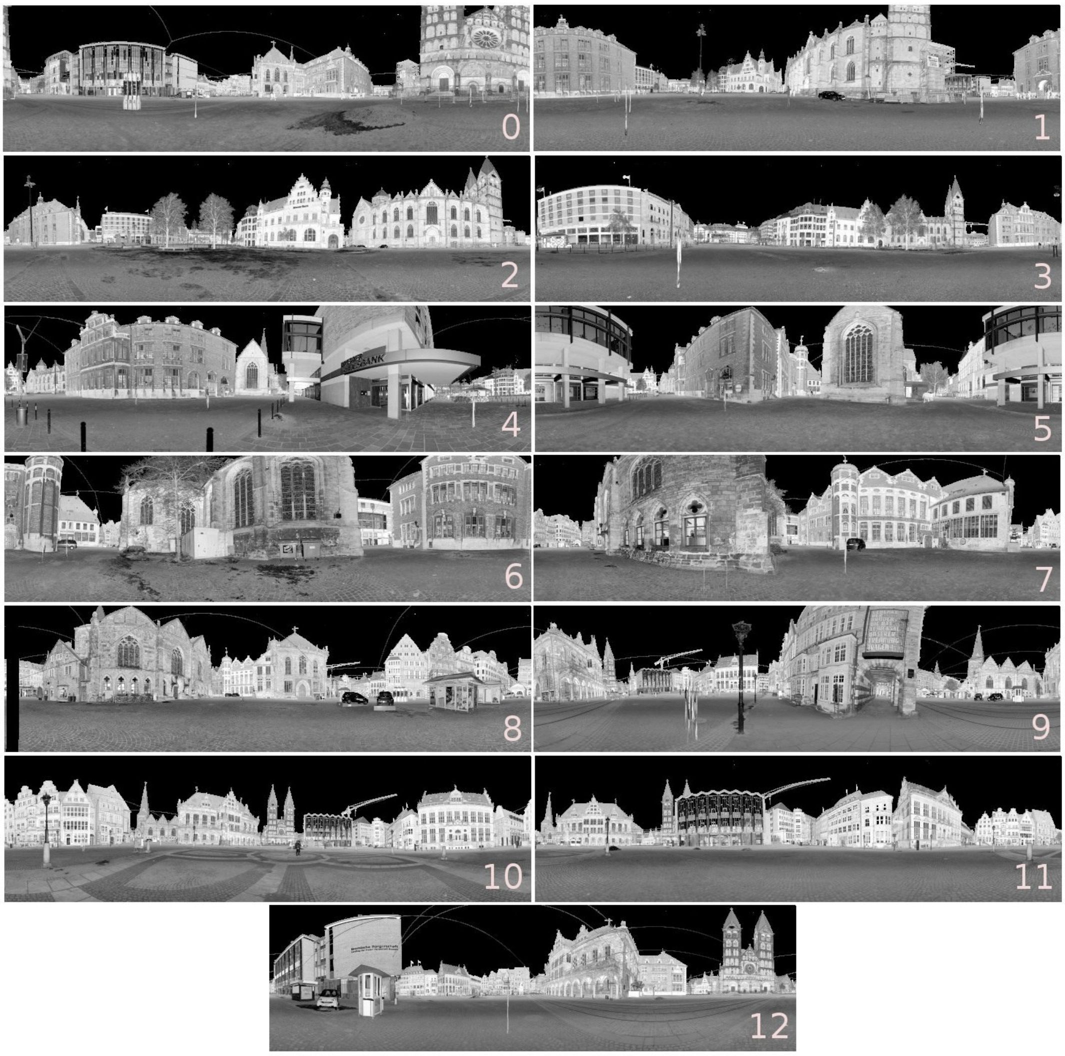


Figure 4.1: Reflectance value panorama images from the scans

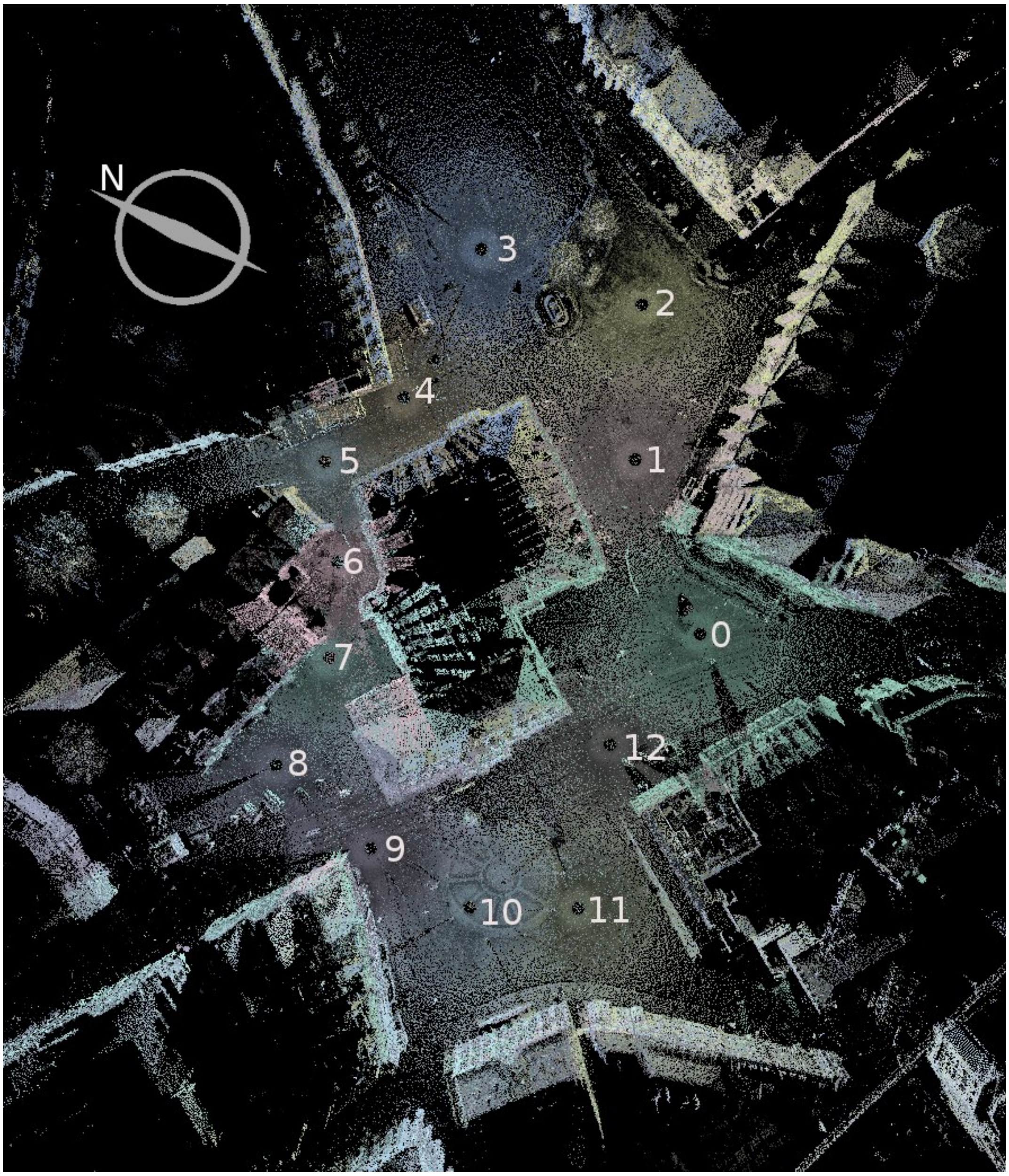


Figure 4.2: A top-down perspective of all 13 registered scans

Table 4.1: Panorama generation times and feature extraction times versus panorama size and point count

Scan id	#points	resol.	pano. gen. time (s)	#feat-ures	feat. ext. time (s)
Scan001	16,164,873	720x200	0.94	2,194	1.41
		1440x400	0.99	9,612	5.67
		2160x600	1.02	22,572	13.54
		2880x800	1.04	41,004	24.43
		3600x1000	1.07	65,606	39.51
Scan003	15,033,238	720x200	0.86	1,914	1.28
		1440x400	0.86	7,286	5.26
		2160x600	0.92	17,588	12.24
		2880x800	0.97	33,547	22.71
		3600x1000	1	55,678	37.01
Scan008	15,854,917	720x200	0.94	2,521	1.41
		1440x400	0.95	10,245	5.79
		2160x600	1	22,667	13.29
		2880x800	1.05	41,191	24.23
		3600x1000	1.07	65,193	39.1
Scan010	15,177,761	720x200	0.89	2,547	1.51
		1440x400	0.93	10,958	6.06
		2160x600	0.96	24,529	13.9
		2880x800	0.99	43,246	25.16
		3600x1000	1.03	66,845	41.66
Scan011	14,607,918	720x200	0.85	2,078	1.36
		1440x400	0.89	8,753	5.7
		2160x600	0.89	20,083	12.88
		2880x800	0.95	35,490	23.59
		3600x1000	0.99	5,5519	37.49
Scan012	15,762,313	720x200	0.91	2,233	1.39
		1440x400	0.93	8,785	5.54
		2160x600	0.94	18,947	12.67
		2880x800	1.03	33,998	22.78
		3600x1000	1.05	53,324	36.9

Table 4.2: Table showing matching and registration results versus panorama size. There are three pairs with large overlap (10-11, 6-7, 2-3), two pairs of small overlap (1-12, 3-4) and one pair with no overlap (3,8)

Scan pair	resolution	#features	#matches	match. time (s)	#inliers	reg. time (s)
	720x200	4625	85	8.96	23	2.08
	1440x400	19711	255	55.47	92	5.24
Scan010 → Scan011	2160x600	44612	497	150.82	142	9.67
	2880x800	78736	784	311.36	223	14.55
	3600x1000	122364	1129	529.95	303	21.72
	720x200	6679	43	13.91	5	0.16
	1440x400	27204	202	82.26	27	4.23
Scan006 → Scan007	2160x600	61677	467	225.99	55	8.8
	2880x800	109059	705	457.88	77	12.56
	3600x1000	169369	1028	765.22	96	18.99
	720x200	4322	55	8.1	9	0.41
	1440x400	17493	161	48.34	33	3.57
Scan002 → Scan003	2160x600	42036	332	143.25	63	6.59
	2880x800	77826	540	304.52	109	10.73
	3600x1000	125314	829	536.88	142	15.45
	720x200	4427	40	8.24	0	/
	1440x400	18397	160	50.53	7	3.4
Scan001 → Scan012	2160x600	41519	272	138.38	26	5.6
	2880x800	75002	493	286.11	38	9.56
	3600x1000	118930	663	522.1	42	13.07
	720x200	4445	37	8.67	4	/
	1440x400	16791	148	45.91	17	3.15
Scan003 → Scan004	2160x600	38774	268	128.03	33	5.46
	2880x800	71735	479	271.94	33	9.03
	3600x1000	116350	651	491.72	57	12.68
	720x200	4435	41	8.49	0	0.13
	1440x400	17531	118	47.53	0	2.78
Scan003 → Scan008	2160x600	40255	255	132.52	0	5.25
	2880x800	74738	379	283.78	0	7.57
	3600x1000	120871	578	507.68	0	10.75

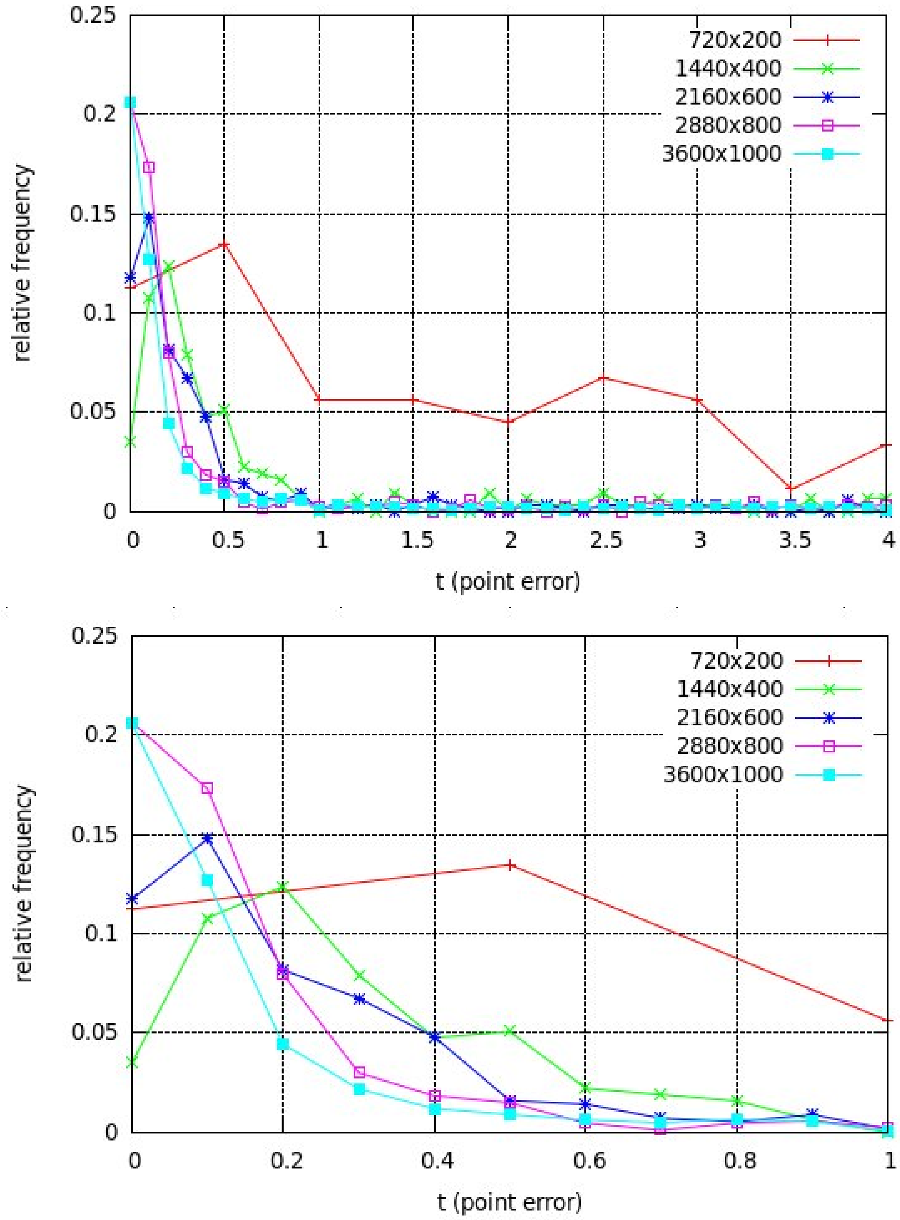


Figure 4.3: Histogram of the error of registration of the points with respect to the best registration. Average of all successful consecutive pairwise registrations

The above histogram gives us a lot of clues. First it helps us determine the threshold parameter t for each resolution. First we can conclude that the $720x200$ is unusable in the sense that it is difficult to distinguish between an inlier and an outlier. For the other resolutions we have decided to choose t as 1.0 for $1440x400$ and $2160x600$, and 0.5 for $2880x800$ and $3600x1000$.

This histogram also gives us a rough estimate of the precision of the registration. We can notice the peak for $1440x400$ is around 0.2, for $2160x600$ is around 0.1 and for the higher resolutions the resolution of the histogram is too low. So we can expect the average of the precision to be around these points.

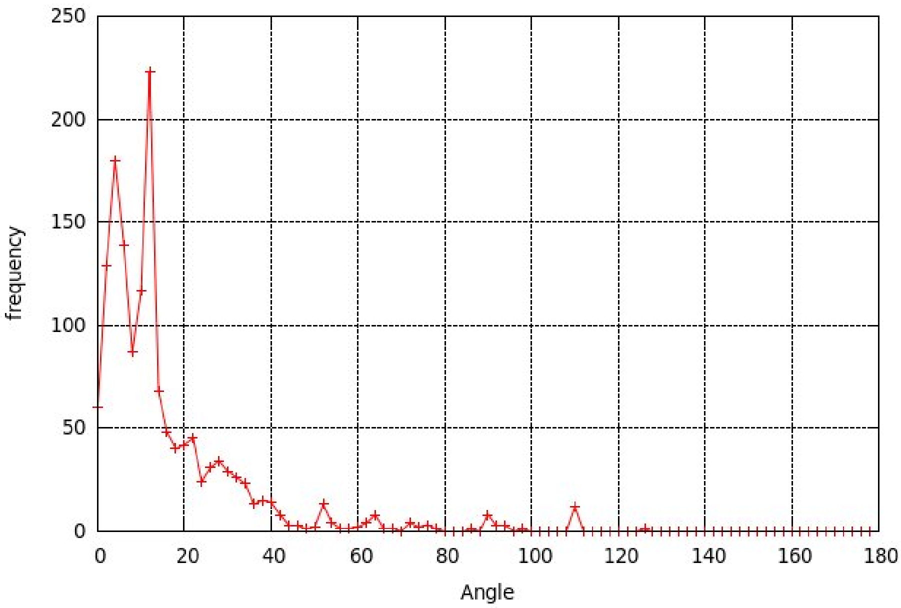
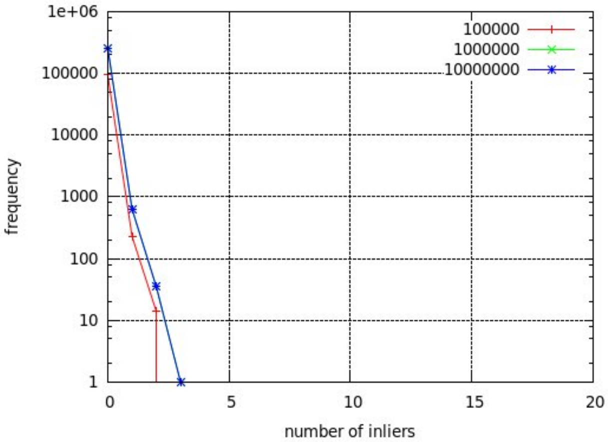


Figure 4.4: Histogram of the angles between the feature points and the positions of the scanner. Average of all consecutive pairwise registrations.

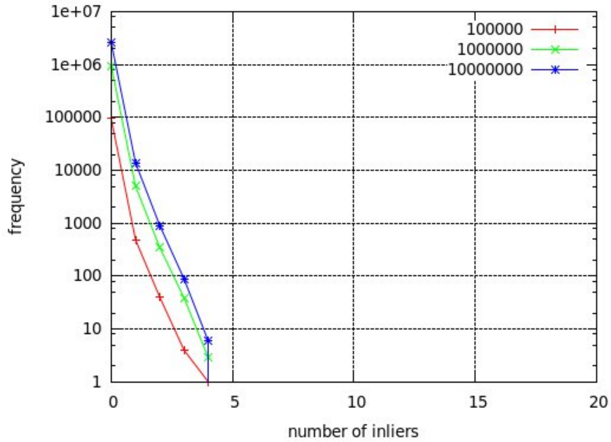
This histogram gives us a good insight on the robustness of the SIFT features. The histogram shows almost no signs of presence of features after 40° and most of the features that the system utilized are under 20°. Thus, we can conclude that this is one significantly big drawback of the system, since in some cases like for example in a street-like environment robustness for larger relative angles is more than necessary.

Table 4.3: Pairwise scan registration results with resolution of 2160x600

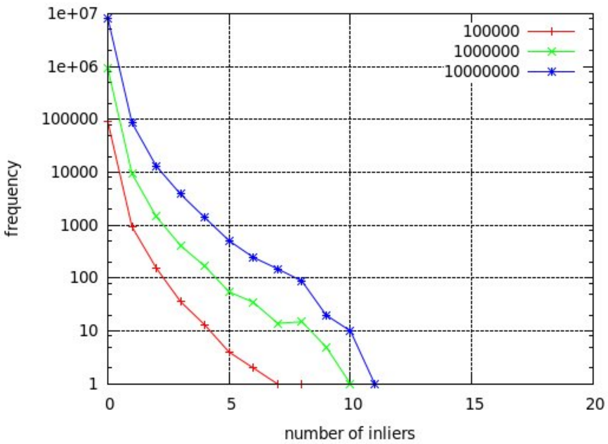
Scan pair	yaw	pitch	roll	x	y	z
Scan000-001	/	/	/	/	/	/
Scan001-002	43.02	1.52	0.59	-23.684	0.7386	24.108
Scan002-003	-24.49	-0.66	0.96	-9.645	0.2869	36.051
Scan003-004	158.54	0.92	3.41	36.554	0.5665	-1.065
Scan004-005	72.68	2.25	1.44	-22.205	0.1318	-0.0574
Scan005-006	-131.54	1.38	-3.43	-21.261	0.00328	-6.088
Scan006-007	-78.68	-6.23	6.92	5.363	-0.1461	20.373
Scan007-008	-51.61	2.43	-1.57	25.268	-0.5506	8.044
Scan008-009	-124.49	-0.76	3.79	-8.622	-0.8420	-26.712
Scan009-010	153.60	2.37	4.76	-11.864	0.8045	22.786
Scan010-011	-79.08	-0.40	-1.38	-12.290	-0.7201	-20.723
Scan011-012	-140.53	0.46	-0.21	4.988	0.1123	37.727
Scan012-000	0.48	0.55	-0.089	26.497	0.2709	-17.144



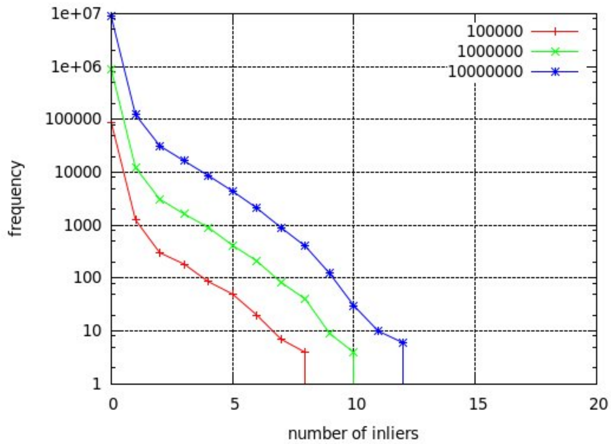
(a) 1440x400 with 118 matches



(b) 2160x600 with 255 matches



(c) 2880x800 with 379 matches



(d) 3600x1000 with 578 matches

Figure 4.5: A figure showing the amount combinations of 3 matches with respect to the number of inliers that the corresponding transformations have. These figures are obtained from the pair of scans 3 and 8 which have no overlap. Also various amounts of iterations are included.

4.1.2 Global registration results

As explained in section 3.4.2 we have implemented a simple technique that offers global registration. The biggest drawback is that it doesn't use any optimization of the results, so errors sum up.

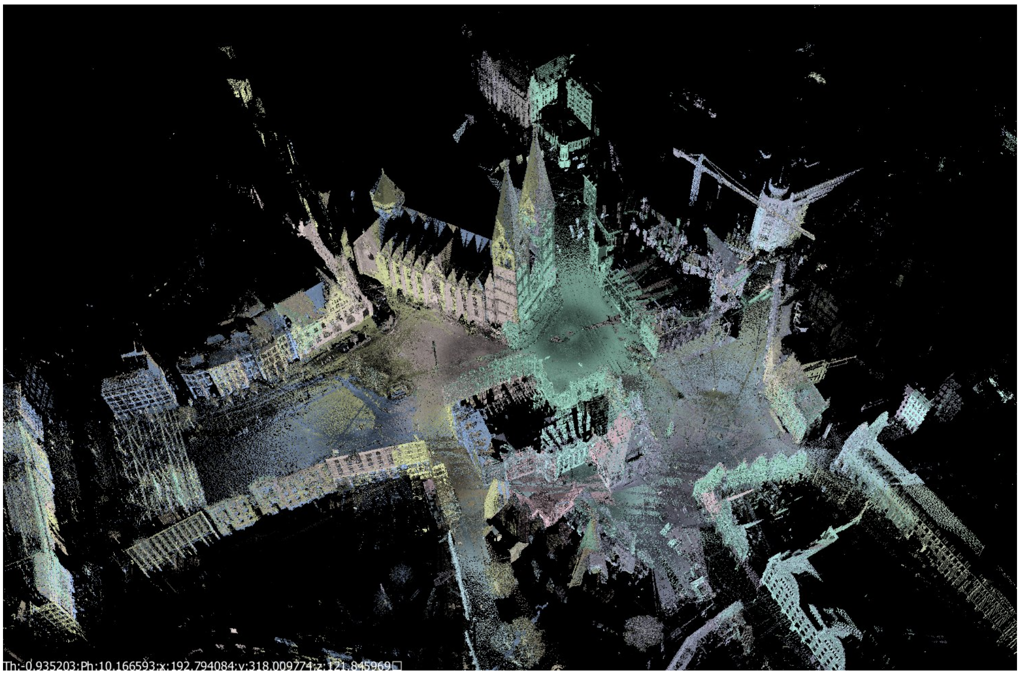


Figure 4.6: A visualization of all 13 registerd scans

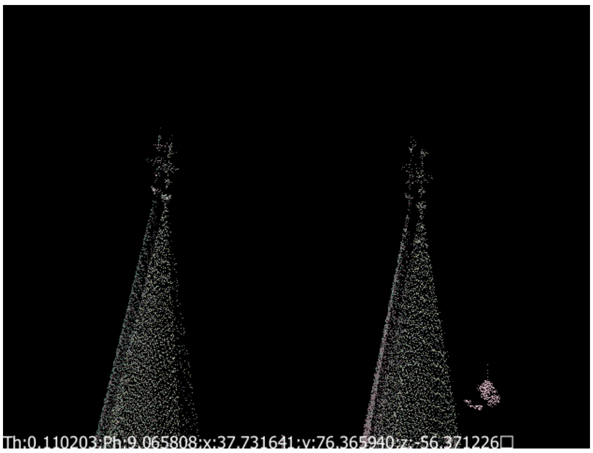


Figure 4.7: The top of the towers. The top of the towers is a very interesting part because it is included by scans in the beginning and in the end of the loop, and the errors of global registration are most visible here. This result has been obtained with a panorama resolution of $3600x1000$

4.2 Street-like dataset

As we have seen from the angle histogram before, the system made most use of features that had relative angles with respect to the scanner's position mostly below 20°. This implies that in some scenarios like for example street-like environments the system will not perform well because it will need to rely on features with larger angles. This holds even for scans with large overlaps.

For this reason we have tested the system on a dataset of a street-like scene. The scans were taken with the half of the resolution of the previous scans, that is 0.08° angular resolution which means that the resolution of the scan is approximately $4500x1250$. The resolution that we have tested these scans is $2160x600$. This resolution is smaller than the theoretical resolution that can be used, primarily due to the restriction of the current implementation of the panorama generation process which can be improved to allow maximum theoretical resolutions.

The panorama images of the scans are available at 4.8, and a visualization of the registered scans is available in figure 4.9

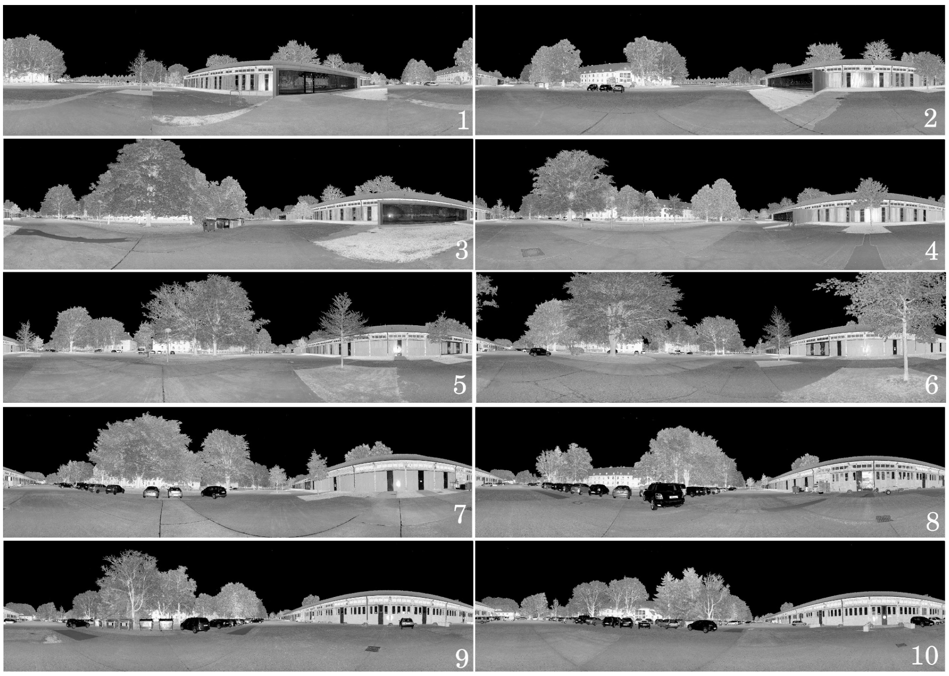


Figure 4.8: Reflectance maps of the 10 scans

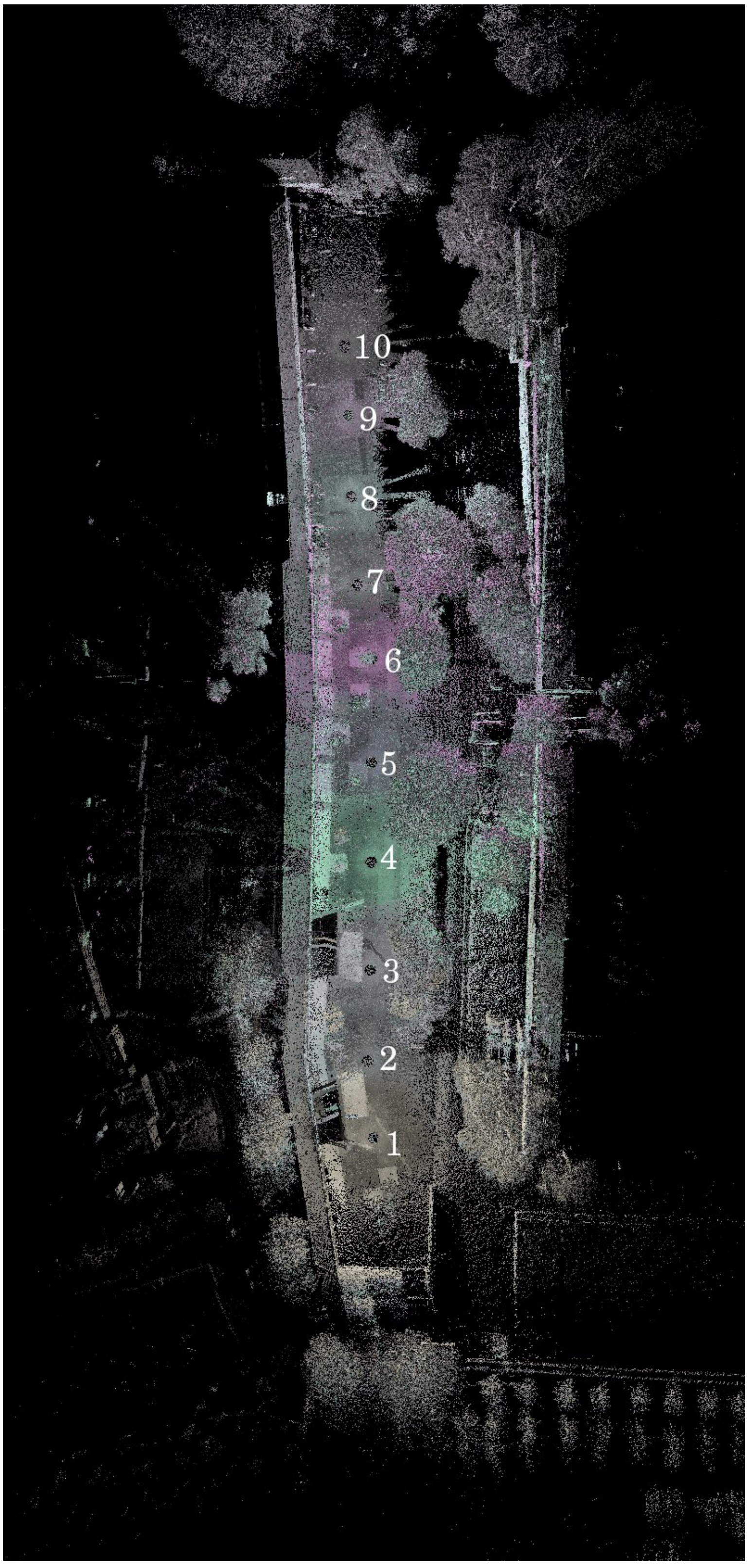


Figure 4.9: Top-down perspective of the positions of the 10 scans

4.2.1 Registration results

Table 4.4: Inlier matches versus total matches for resolution of 2160x600

Scans	#inl.	#tot.	Scans	#inl.	#tot.
$Scan001 \rightarrow Scan002$	5	221	$Scan006 \rightarrow Scan007$	11	237
$Scan002 \rightarrow Scan003$	5	204	$Scan007 \rightarrow Scan008$	7	254
$Scan003 \rightarrow Scan004$	5	227	$Scan008 \rightarrow Scan009$	13	234
$Scan004 \rightarrow Scan005$	6	222	$Scan009 \rightarrow Scan010$	23	284
$Scan005 \rightarrow Scan006$	9	248			

As we can see our expectations were correct and registration with this method in street-like environments is indeed more difficult. We can see that from Scan006 to Scan010 the amount of inliers was higher, and these numbers were satisfactory for a successful registration. On the other hand, for the scans from Scan001 to Scan005 the amount of the inliers was very low, and required allowing a successful registration only for 2 inliers per registration, as the other 3 are used to obtain the transformation. This value is to low as the probability to obtain false positive result with 2 inliers is quite high, and in this dataset this proved to be the case, so we had to manually adjust the system to exclude outliers so that we obtain the result we have.

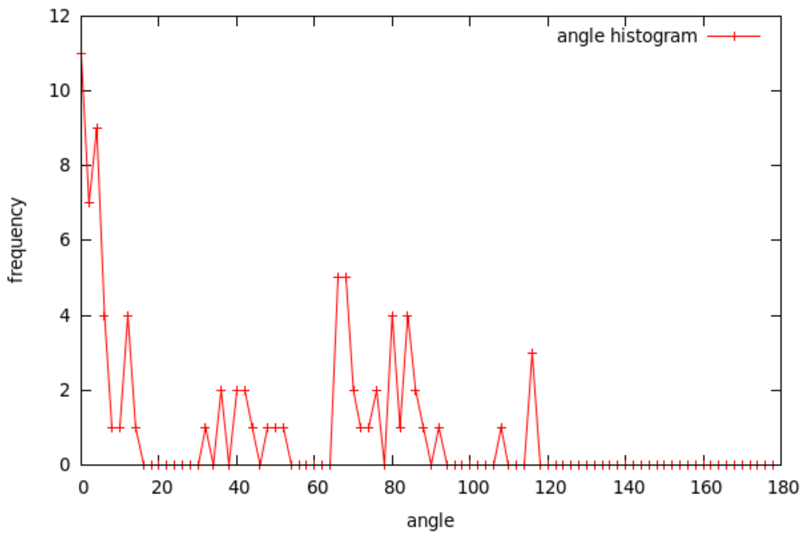


Figure 4.10: Angle histogram for the street-like dataset

As we can see from the angle histogram in this case, the amount of features with higher angles increased however there is still a significant amount of features of low angles, so we can see this is the decisive factor in the failure of the system with this dataset.

We can notice that the robustness of the SIFT features is lower than that presented in the original paper [Low04], and we can argue that a reason for this maybe that we are using different kind of projection of the scene. Namely we are using an equirectangular projection, while most probably the method has been tested on rectilinear projection which is mostly used in conventional photographic lenses. Since the distortions involved in these two types of projections are different, the robustness of the SIFT method in this respect obviously varies.

5

Future work

5.1 Method improvements

One thing that will drastically improve the inlier to outlier ratio is to filter matches during registration phase not only by the correspondence of the triangles but also taking into consideration the feature's scale and orientation. Since the scale directly depends on distance and the orientation on the relative position of the scans and on the feature's position, this can be combined with the current rotation and translation estimations to decide whether some 3 point pair matches are an inlier registration or not.

One of the most important things however, is to implement a more sophisticated method for global registration. This might be only some iterative optimization that will only work on the set of inliers, or a more complex method which will also find false positive registrations and discard them.

Another thing is that the relative angle for the features is also dependent on the type of projection. Currently we are only using the most basic equirectangular projection, and it may be the case that the efficiency of this part of the system can be improved by using some other not so trivial projection. This, however, must go through series of tests first.

Another possible improvement to the system would be to develop a method which will decide if a registration is successful based on finding peaks in the histogram of amount of inliers as described in 4.1.1. The possible improvement however is also not certain for this as it may also be possible to achieve this effect by testing for the best possible d value which will create the best trade-off between number of false positive and the number of disregarded true positives.

Also one additional improvement might be to test the registration results on the point clouds themselves. This will break the positive part of the current method which is that it doesn't involve the point cloud in the registration process however it will certainly provide better results.

Another idea for the use of the SIFT method for registration might be to extract and match SIFT features not from panorama images from the perspective of the scanner, but to work on images that are a vertical orthographic parallel projection of the point clouds. This will provide only 2D transformation parameters, however these might be very useful in some cases, like testing for false positives.

One small improvement should be made to the current approach of generating panorama images. Namely, currently when the system finds correspondences between the points in the point cloud and the super-pixels in the panorama object, it considers only one super-pixel and ideally the system should consider four super-pixels and fill their values accordingly.

5.2 Additional tests

Additional series of tests could be made for testing the effect of histogram equalization on the feature extraction and matching process. In that case also different types of histogram equalizations can be tested.

Another extension of the current tests could be providing better tests for the precision of the system on various resolutions.

6

Conclusion

Registration of terrestrial laser scans is without a doubt a very interesting field in the scientific world. Many approaches are offered but every approach has its own weaknesses. The registration method based on SIFT features on reflectance images is no exception to that statement. We have seen in our test results that the major weakness of this method is that it works on features that have relative angle of mostly up to $20°$ between the features and the scanner's positions, which is quite constrictive. For example for the environments like hallways or streets where larger angles are crucial the method will have a hard time finding inlier features.

Apart from that the method showed very good results in different aspects. Precision was very good even in low resolutions, and even though in the whole process of registration the system had no contact with the large point cloud data and it only relied on low resolution data maps the end result was very close to the actual values. This gives the big advantage that instead of working on extremely large files which in our case were point clouds of around 15,000,000 points, the system performed well even with resolutions of $1440x400$ and $2160x600$ which is about 10 to 30 times less data.

Performance wise the conclusion is that it depends on the resolution. The biggest slowdown for the system presented the matching of the SIFT features. All other steps took a small fraction of the time needed to match features which ranged from around 50s for a resolution $1440x400$ to around 500s for a resolution of $3600x1000$. This is of course expected as the system needs to check for matches in amounts of 100,000 features all of which hold very high dimensional description vectors of 128 dimensions.

Finally it can be concluded that the method presents one intuitive aspect of registration of point clouds. Since it works on features that depend on the type, color and other attributes of the surfaces and not on the geometrical arrangements that the point clouds form it can be characterized as a complement of the methods which utilize the latter attribute like for example the method of registration based on planar patches. This means that it can be combined with these other methods in the same way that all the methods are combined with the ICP algorithm to give one better system.

Bibliography

[BB07] Jan Böhm and Susanne Becker, *Automatic marker-free registration of terrestrial laser scans using reflectance features*, In: Proceedings of 8th Conference on Optical 3D Measurment Techniques, 2007, pp. 338–334.

[BDW+04] Gerhard H. Bendels, Patrick Degener, Roland Wahl, Marcel Körtgen, and Reinhard Klein, *Image-based registration of 3d-range data using feature surface elements*, The 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST 2004) (Y. Chrysanthou, K. Cain, N. Silberman, and Franco Niccolucci, eds.), Eurographics, December 2004, pp. 115–124.

[BF07] Shahar Barnea and Sagi Filin, *Registration of terrestrial laser scans via image based features*, In: ISPRS Workshop on Laser Scanning, 2007.

[BL97] Jeffrey S. Beis and David G. Lowe, *Shape indexing using approximate nearest-neighbour search in high-dimensional spaces*, Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **0** (1997), 1000.

[BM92] Paul J. Besl and Neil D. McKay, *A method for Registration of 3–D Shapes*, IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992), no. 2, 239 – 256.

[DB06] Christoph Dold and Claus Brenner, *Registration of terrestrial laser scanning data using planar patches and image data*, In: International Archives of Photogrammetry and Remote Sensing, 2006, pp. 25–27.

[Dol05] Christoph Dold, *Extended gaussian images for the registration of terrestrial scan data*, 2005.

[FB81] Martin A. Fischler and Robert C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Commun. ACM **24** (1981), no. 6, 381–395.

[GA05] Armin Gruen and Devrim Akca, *Least squares 3d surface and curve matching*, ISPRS Journal of Photogrammetry and Remote Sensing **59** (2005), no. 3, 151–174.

[GRB94] Guy Godin, Marc Rioux, and Rejean Baribeau, *Three-dimensional registration using range and intensity information*, vol. 2350, SPIE, 1994, pp. 279–290.

[hBL04] Kwang ho Bae and D. D. Lichti, *Automated registration of unorganised point clouds from terrestrial laser scanners*, In: International Archives of Photogrammetry and Remote Sensing, Vol. XXXV, Part B5, Proceedings of the ISPRS working group V/2, 2004, pp. 222–227.

[Hor84] Berthold K. P. Horn, *Extended gaussian images*, Proceedings of the IEEE **72** (1984), no. 2, 1671–1686.

[Hor87] ———, *Closed-form solution of absolute orientation using unit quaternions*, Journal of the Optical Society of America A **4** (1987), 629–642.

[KLZ+09] Zhizhong Kang, Jonathan Li, Liqiang Zhang, Qile Zhao, and Sisi Zlatanova, *Automatic registration of terrestrial laser scanning point clouds using panoramic reflectance images*, Sensors **9** (2009), no. 4, 2621–2646.

[Low99] David G. Lowe, *Object recognition from local scale-invariant features*, Computer Vision, IEEE International Conference on **2** (1999), 1150.

[Low04] ———, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision **60** (2004), no. 2, 91–110.

[MPD06] Ameesh Makadia, Alexander IV Patterson, and Kostas Daniilidis, *Fully automatic registration of 3d point clouds*, CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Washington, DC, USA), IEEE Computer Society, 2006, pp. 1297–1304.

[MS05] Krystian Mikolajczyk and Cordelia Schmid, *A performance evaluation of local descriptors*, IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005), 1615–1630.

[N0̈9] Andreas Nüchter, *3d robotic mapping - the simultaneous localization and mapping problem with six degrees of freedom*, Springer Berlin / Heidelberg, 2009.

[Now05] Sebastian Nowozin, *autopano-sift*, 2005, `http://user.cs.tu-berlin.de/~nowozin/autopano-sift/`.

[Rie09] Riegl, *RIEGL VZ-400*, September 2009, http://www.riegl.com/uploads/tx_pxpriegldownloads/10_DataSheet_VZ400_10-09-2009_01.pdf.

[SB05] Sara Schuhmacher and Jan Böhm, *Georeferencing of terrestrial laserscanner data for applications in architectural modelling.*

[TAB+04] J. Talaya, R. Alamus, E. Bosch, A. Serra, W. Kornus, and A. Baron, *Integration of a terrestrial laser scanner with gps/imu orientation sensors*, ISPRS Congress Istanbul, 2004.

[WB08] Zhi Wang and Claus Brenner, *Point based registration of terrestrial laser data using intensity and geometry features*, ISPRS Congress Beijing, 2008.

Proclamation

Hereby I confirm that I wrote this thesis independently and that I have not made use of any other resources or means than those indicated.

Bremen, May 2010