3D room mapping with a hand-held ToF camera
Technical report

Tobias Grundström
tobgr602@student.liu.se

Anna Hjelmberg
annhj876@student.liu.se

Mats Nilsson
matni403@student.liu.se

Fredrik Olsson
freol454@student.liu.se

2017-12-15

Contents

1 Introduction

This project is about delivering a program which utilizes the Fotonic G-series depth camera to produce a 3D model of an indoor environment.

1.1 Problem description

When capturing point clouds, every single scan will produce a point cloud which will have its own coordinate system, this system will differ from the global (reference) coordinate system, based on the location and direction of the camera. The problem of finding a transformation matrix which will transform all the point-clouds to the common global coordinate system is called *Point Cloud Registration*, which is the problem that will be dealt with in this work.

1.2 Project Overview

The goal of this project is to develop a program that can produce a 3D model of a room from a sequence of Time-of-Flight data. The data should be collected using a Fotonic G-series camera which produces depth images. The project is divided into three main parts; point-cloud logging from sensor, registration and outlier filtering of the point clouds. On top of this, a mesh of the resulting model should be used to visualize the results. An overview of the system is described in further detail in subsection 4.4.

1.3 Client

The client for this project is Per-Erik Forssén, associate professor at the Computer Vision Laboratory at the Department of Electrical Engineering at Linköping University, in association with the company Fotonic.

2 Related Work

The depth map registration problem has been handled by several previous works. One recent work is the Fast Global Registration [1] (FGR), that aims at creating a fast and robust solution to the registration problem. This is done by estimating a rigid transformation from one point cloud to the other. Firstly by using two verification steps to find the best correspondences in the data set and secondly by finding the transformation that aligns these point clouds. Also a frequently used method is the Generalized Iterative Closest Point [2], GICP, which aims to minimize the difference between two point clouds by finding the closest matching points and estimating the transformation from them and iterating until the solution has converged. Another work that has been taken into

account is the Probabilistic Framework for Color-Based Point Set Registration [3], that includes color features and Gaussian mixture models to solve the registration problem globally.

3 Theoretical background

In the following section, a short introduction to the theoretical background of the methods used in the project will be presented.

3.1 Fast Point Feature Histograms

The Fast Point Feature Histogram, FPFH, is an optimized version of the Point Feature Histogram, PFH, using a revised theoretical formulation to reduce computational times. Here the FPFH will be presented, for further reading on the PFH see [4].

The Fast Point Feature Histogram at point p is calculated using an estimated surface normal at the point and a set of neighboring k points. The advantage of the FPFH in 3D registration is its invariance to rigid transformations while being insensitive to point cloud density and small amounts of noise [5]. The computation of FPFH is performed as follows:

1. For every pair of points $\mathbf{p_i}$ and $\mathbf{p_j}$ in the set of neighbors $\mathcal{P}^k$ and the corresponding normals $\mathbf{n_i}$ and $\mathbf{n_j}$, compute the angular variations of the normals using

$$\begin{aligned} \alpha &= v \cdot \mathbf{n_j}, \\ \phi &= (u \cdot (\mathbf{p_j} - \mathbf{p_i}))/\|\|\mathbf{p_j} - \mathbf{p_i}\|\|, \\ \theta &= \arctan(w \cdot \mathbf{n_j}, u \cdot \mathbf{n_j}). \end{aligned} \tag{1}$$

   Where $u = \mathbf{n_i}$, $v = (\mathbf{p_j} - \mathbf{p_i}) \times u$ and $w = u \times v$.

2. For all sets of α, ϕ and θ a histogram called Simplified Point Feature Histogram, SPFH, is created. By also taking all the neighbors' SPFH values into account the FPFH is calculated as:

$$FPFH(\mathbf{p_i}) = SPFH(\mathbf{p_i}) + \frac{1}{k} \sum_{i=1}^{k} \frac{1}{\omega_k} \cdot SPFH(\mathbf{p_k}). \tag{2}$$

   Where the weight w_k is chosen as the distance between the point $\mathbf{p_i}$ and the point $\mathbf{p_k}$. [5]

3.2 Fast Global Registration

The implementation made by [1] uses the pairwise registration of Fast Global Registration that performs registration by sequentially adding several pairwise aligned point clouds.

The algorithm takes the calculated FPFH features and point clouds corresponding to each depth image, called $\mathbf{P}$ and $\mathbf{Q}$, as input and estimates the transformation that aligns $\mathbf{Q}$ to $\mathbf{P}$. By performing two types of tests on the sets of correspondences between the two clouds, some points are disregarded. This leads to increased inlier ratio and reduced computational times. The first test is the reciprocity test that only chooses a correspondence pair $(\mathbf{p}, \mathbf{q})$ if and only if $FPFH(\mathbf{p})$ is the nearest neighbor to $FPFH(\mathbf{q})$ of the entire set of $FPFH(\mathbf{P})$ and if $FPFH(\mathbf{q})$ is the nearest neighbor to $FPFH(\mathbf{p})$ of the set of $FPFH(\mathbf{Q})$. The second test is called the tuple test and is performed by randomly picking three correspondence pairs $(\mathbf{p_1}, \mathbf{q_1})$,$(\mathbf{p_2}, \mathbf{q_2})$ and $(\mathbf{p_3}, \mathbf{q_3})$ and if they fulfill the constraint in equation 3 they are said to be compatible and will be used in the registration. For these points, the aligning transformation is optimized over several iterations by minimizing the sum of the distances between the points $\mathbf{p}$ and the transformed points $\mathbf{Tq}$ where $\mathbf{T}$ is the estimated rigid transformation. [1]

$$\tau < \frac{||\mathbf{p_i} - \mathbf{p_j}||}{||\mathbf{q_i} - \mathbf{q_j}||} < 1/\tau, \qquad \forall i \neq j. \tag{3}$$

3.3 Iterative Closest Point Registration

To refine the transformation solution from the Fast Global Registration a few iterations of Iterative Closest Point Registration, ICP, is used. In the project a version of ICP called Generalized ICP, GICP, as described in [2] is used. A common way to describe the ICP algorithm is:

1. Find the closest match in the target point cloud for every point in the source cloud
2. Estimate a transformation by using a point to point root mean square approach
3. Transform points
4. If not converged return to step 1 and iterate again

In the standard ICP-algorithm, the minimization step (estimating the transformation) is done as

$$T \leftarrow \underset{T}{\operatorname{argmin}}\{\sum_i w_i ||T \cdot b_i - a_i||^2\}$$

where T is the proposed transform, b_i a point in the source cloud that correspond to the point a_i in the target cloud. In the Generalized ICP, a probabilistic model is attached

to the minimization step yielding a new minimization modeled as

$$T \leftarrow \underset{T}{\operatorname{argmin}} \sum_i d_i^{T'} (C_i^B + T C_i^A T')^{-1} d_i^T$$

where T is again the proposed transform, $d_i^{T'} = b_i - T a_i$ and C_i^X are covariance matrices associated with the measured points. This is under the assumption that we have removed all the correspondences with a distance which is larger than a threshold d_{max}. From the above the standard ICP algorithm can be extracted by setting $C_i^B = I$ and $C_i^A = 0$.

3.4 Multipath Interference

The camera used in the project is a Time of Flight (ToF) camera that measures an infrared light pulse that travels to an object and then is reflected back to the camera. Using this method for measuring distance opens up for a specific type of measuring error known as *multipath interference*. The problem of multipath interference arises when measuring concavities, for example when the camera is directed against a corner between two walls as depicted in Figure 1. The camera is based on measuring the reflection of an infrared pulse and since most surfaces are not perfect reflectors, some light is partially scattered. Any scattered light that reflects back into the sensor will add to measurement and thus depict the distance to a point as further away then it actually is. This typically gives sharp corners a smoother more blob-like appearance as can be seen in Figure 1.

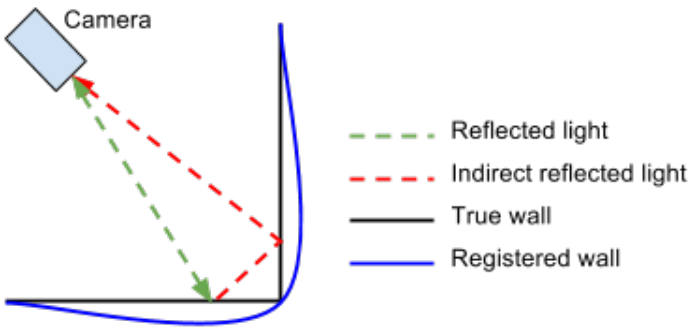

Figure 1: Multipath interference occurs when indirectly reflected light is measured

3.5 Radial Distortion

Radial distortion is a type of optic distortion that originates from the use of a lens. It introduces curvature where there should be none. This is clearly highlighted in Figure 2, which shows a set of orthogonal equispacial lines subject to barrel distortion.

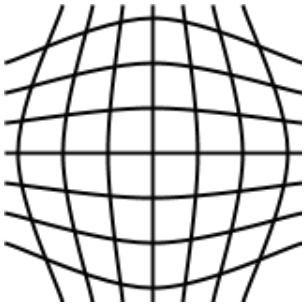

Figure 2: A radially distorted orthogonal grid

3.6 Reflectivity Characteristics

When capturing images there are mainly three factors that affect the amount of light an object reflects that will be registered by the camera. Those factors are reflectivity coefficients, angles of incidence and diffuse or direct reflections.

The reflectivity coefficient is a measurement of how much light an object reflects. The reflectivity coefficient varies between different materials but is also different for different wavelengths of light. Objects which appears light or dark to the human eye can behave differently when observed in for example near infrared light which the Fotonic G-series camera uses (around 850 nm).

Light scatters from a surface in different ways depending on the material. Materials with a diffuse reflection are easier to capture with a time-of-flight camera than materials with a direct reflection, for example a mirror. Also the angle of incident the light hits a surface will matter when capturing images [6].

With this in mind, some scenes will be more difficult than others to reconstruct. For example, in the *furniture scene* (Figure 6b) described later in this report, there is a green wall which is not at all visible in the depth images.

4 Method

In this section, the pipeline in its entirety will be presented. It will contain all the before mentioned methods in Section 3 as well as some other design choices made to solve the registration problem.

4.1 Voxel Grid Filtering

Downsampling of point clouds is performed using a voxel (volumetric pixel) grid filter. The size of the voxels in the grid specifies the rate of downsampling, where a large voxel is equivalent to a large downsampling and vice versa. The voxels are positioned side-by-side in the point cloud and all points contained in the voxel are used to calculate the centroid value, which is then used as the new estimated data point while all other points are removed [7].

4.2 Removal of Image Noise

When capturing images there is always some sort of noise in the image. A specific kind of noise, which appears in depth images captured with time-of-flight technology, is the so called 'flying pixels' at depth edges. These appear when a sensor captures reflecting signals from edges of a target object in the scene. The reflected light in the sensor is registered both from the edge of the object and from the background. These artifacts result in errors created by interpolated values between the measured object and the background. [8] Figure 3 shows an unfiltered image from one of the datasets used in this project. The figure shows points of target objects and points which are noise.

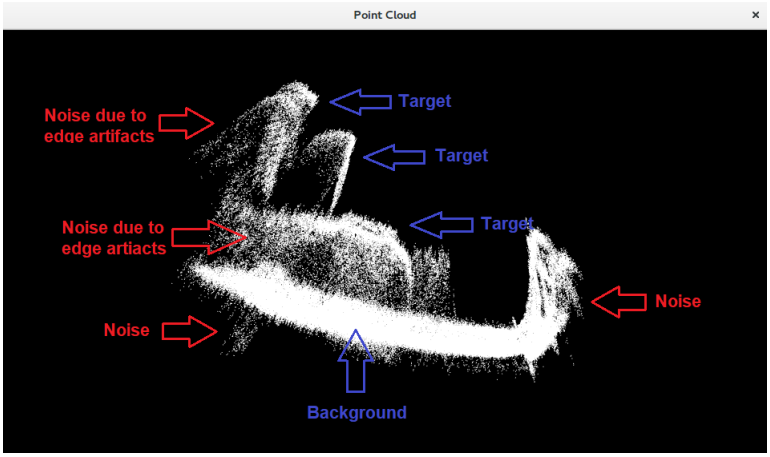

Figure 3: Point cloud with targets and noise.

In this project points will be marked as outliers and removed using the PCL function *Statistical Outlier Removal* in the following way:

1. The mean distance of the **k** closest neighbors to the query point is calculated.
2. Points with a distance larger than σ standard deviation of the mean distance to the query point will be marked as outliers.

4.3 Normal Estimation

Normal estimation for every point is performed using least-square plane fitting, where the plane is represented by a point and a normal. To solve this the point $\mathbf{x}$ is calculated as the centroid of the point set of k neighboring points as in equation (4), $\mathcal{P}^k$, and the normal $\tilde{\mathbf{n}}$ is estimated using Principal Component Analysis, by performing Eigenvalue Decomposition on the point set covariance matrix in equation (5).

$$\mathbf{x} = \frac{1}{k} \sum_{i=1}^{k} p_i \tag{4}$$

$$\mathcal{C} = \frac{1}{k} \sum_{i=1}^{k} w_k \cdot (\mathbf{p_i} - \mathbf{x}) \cdot (\mathbf{p_i} - \mathbf{x})^T \tag{5}$$

If the eigenvalues λ_j fulfill the requirement of $0 \leq \lambda_0 \leq \lambda_1 \leq \lambda_2$ the eigenvector corresponding to the smallest eigenvalue λ_0 is an approximation of the normal $\vec{n}$ with ambiguous orientation. The ambiguity can be solved by knowing the viewpoint of the data set $\mathbf{v}_p$ and to change the sign of all normals that do not fulfill equation (6). [9]

$$\vec{n}_i \cdot (\mathbf{v}_p - \mathbf{p}_i) > 0 \tag{6}$$

4.4 System Design Overview

Figure 4 is an overview of the system design illustrated in a flowchart.

The data is generated using the Fotonic G-series camera and extracted using the Fotonic API. Using the API the images are converted to point clouds and passed on to the pre-processing block. To increase computation speeds of several methods the first pre-processing action is the *voxel grid filtering* (Section 4.1) to reduce the number of data points followed by the *statistical outlier removal* (Section 4.2) to remove redundant points and further increase inlier rate. The final step of the pre-processing is *normal estimation* using the method from Section 4.3 which then is used to generate the *Fast Point Feature Histograms* mentioned in section 3.1.

The down-sampled points and corresponding point features for all pairs of point clouds are passed on to the *Fast Global Registration* algorithm'(Section3.2) which gives a good initial solution to the aligning transformation. By running the *Generalized Iterative*

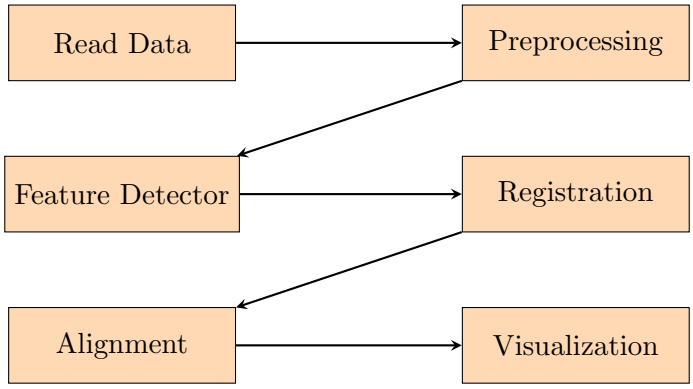

Figure 4: Overview of the system that shows how data is passed between modules.

Closest Point algorithm (Section 3.3) with the previous output translation as an initial guess the solution is refined even further. All point clouds are aligned to the same coordinate system using the estimated transformations. To finalize the process a second down-sampling using the *voxel grid filter* is performed since there will be multiples of the same points in some places when the point clouds are combined. To visualize the final set of points a new set of corresponding surface normals are computed on the combined point cloud. Based on these normals the mesh grid surfaces are estimated and rendered together with the points.

5 Results

In this section the results of the project will be presented. The results presented consist of both quantitative and qualitative results in the form of tables and figures. Six different scenes have been tested and the results of the tests are presented in Table 2. The images in Figure 6 show the different scenes together with the name of which it will be referred to in this text. In Table 2, the mean number of matching correspondences in FGR for all image pairs are presented, meaning how many matches are found in each registration of an image pair. The maximum number of allowed correspondences is set to 3000 to avoid a large decrease in registration speed. This means that a mean number of correspondences close to 3000 is a good result. The second metric is the Root Mean Square (RMS) error of the distances between corresponding points from all image pairs, given in millimeters (mm). Here an RMS of 0 is the best possible result.

Data was gathered by taking a series of images of either 6 or 8 images with high overlap at different indoor locations using the Fotonic G-series camera. For every data set two tests were performed, firstly by using all available images and secondly by using every second image in the sequence. In Table 1 all available parameters and chosen values during testing are showed.

Table 1: Parameter table

Parameter	**Value**
Outlier filtering: k	50
Outlier filtering: σ	1
Normal estimation: Radius	300
FPFH estimation: k	50
FGR: Max corr.	3000

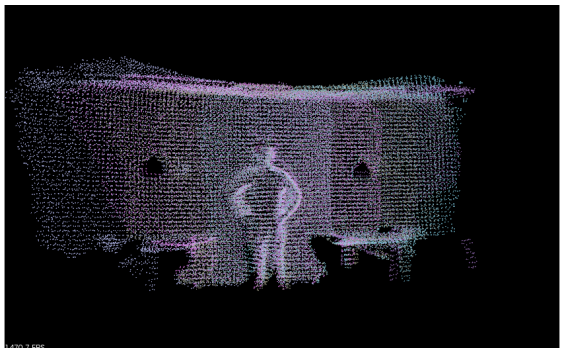

Figure 5: An example of a registration from a sequence of 6 images. This example is referred to as *Kitchen 2 scene*

Table 2: Results of running the algorithm on all the data sets. Shows number of images used, number of matches and Root mean square distance in milimetres.

Dataset	**No. images**	**Matches**	**RMS**
Corridor	8	1243.28	429.79
Corridor	4	108	802.43
Furniture	6	393	632.21
Furniture	3	13.5	681.66
Kitchen 1	6	1998	346.22
Kitchen 1	3	474	625.91
Kitchen 2	6	2884.2	130.96
Kitchen 2	3	2347.5	184.63
Printer	6	2374.8	200.46
Printer	3	123	529.28
Door	6	714.6	303.15
Door	3	28.5	639.31

(a) The corridor scene

(b) The furniture scene

(c) The first kitchen scene

(d) The second kitchen scene

(e) The printer scene

(f) The door scene

Figure 6: Images from the different scenes which have been tested. The name of each scene is written below each image.

5.1 Mesh

From the final set of points, where all images are aligned, new surface normals are estimated and used to generate a grid mesh of the scene to connect neighboring surfaces instead of visualizing points. The mesh was generated through a greedy triangulation algorithm. A resulting point cloud and corresponding grid mesh are shown in figure 7 and figure 8 respectively.

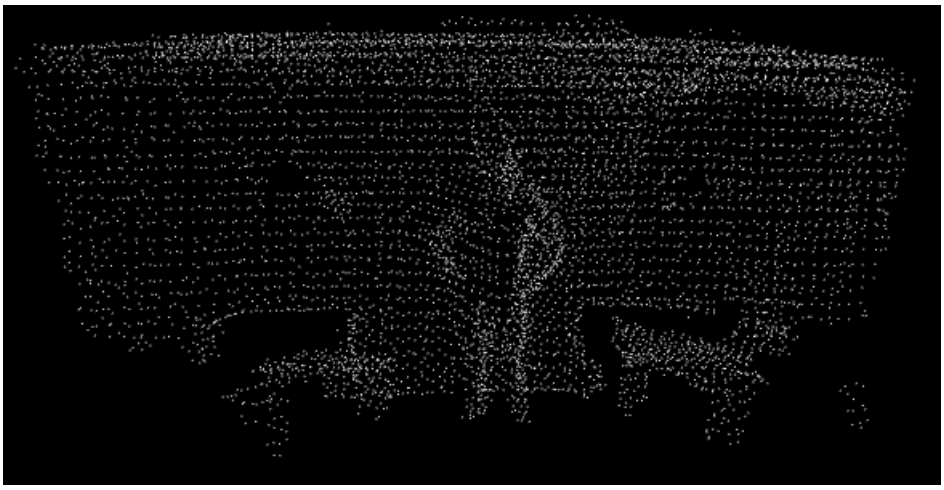

Figure 7: All point clouds in the same image after alignment of the Kitchen 2 scene

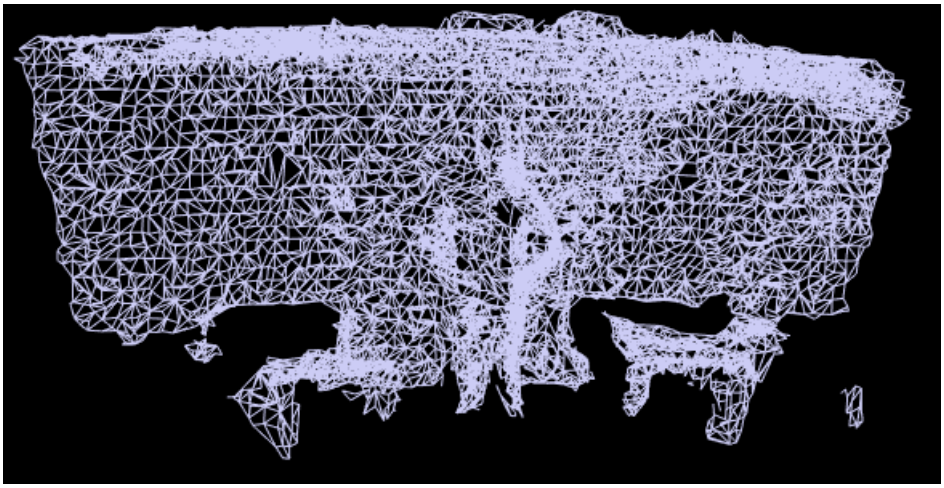

Figure 8: Generated mesh grid from the aligned point clouds of the Kitchen 2 scene

6 Discussion

This section contains a discussion of the results, focusing on challenges, what could be done to overcome them and some interesting points for future work.

6.1 Multipath Interference

The dataset captured and used in the project is often subject to multipath interference, a type of error in the captured images, which is described in Section 3.4. An example is given in Figure 9, which comes from the dataset Printer.

Even though multipath interference is common in room mapping since a room more or less consists of sharp corners and there are some suggestions for working around this presented in [10], it was found to be beyond the scope of this project to tackle this problem. It should, however, be mentioned that this would be an interesting continuation of the project.

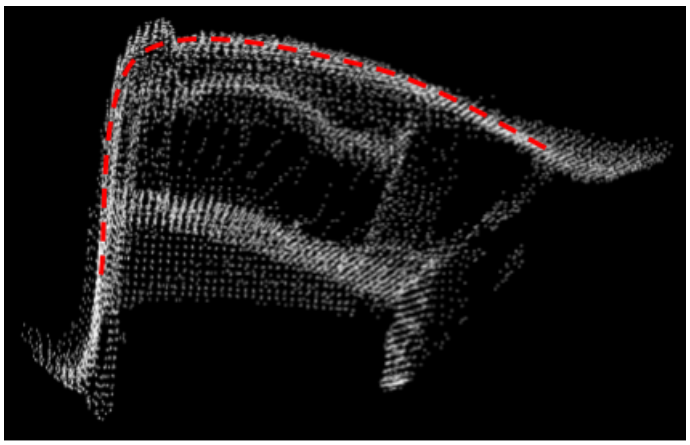

Figure 9: Example of multipath interference

6.2 Radial Distortion

The camera used for capturing the datasets had no easily accessible on-camera rectification processing thus leaving the captured images subject to radial distortion (explained in Section 3.5). Any effect on the captured point clouds reasonably effects the registration process, and thus radial distortion could explain the curvature of the straight back wall as seen in Figure 10.

There exist solutions for rectifying the images which would reduce the effect of radial distortion but it was found to be outside the scope of this project to implement it.

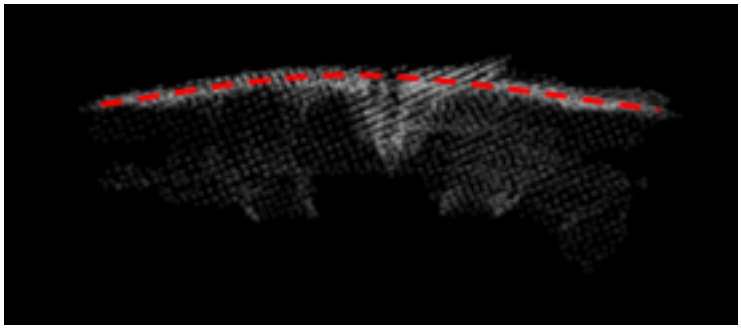

Figure 10: Radial distortion

It should nevertheless be mentioned that this would be an interesting addition to the project.

6.3 Removal of Flying Pixels

As explained in Section 4.2, noise from edge artifacts should be removed before any calculations are made on the image. This should, in theory, make the registration better since all the 'fake points' will be removed from the images resulting in better and correct correspondences. In practice, we have noticed that this is not the case. When removing all (most) outlier in the image the registration did not result in an accurate 3D point cloud.

We have come to the conclusion that this is due to the correspondences which are found when performing FGR. Figures 11-14 show two images of the same scene, from two different angles. Figure 11 and 12 illustrate unfiltered point clouds with the corresponding points marked with white dots. Figure 13 and 14 illustrate the same images but with filtered point clouds. We see that many of the corresponding points in the unfiltered case are points on the human standing in the image and at the edge artifacts. When filtering the images, removing the edge artifacts, the FGR algorithm does not find as many correspondences as before, which makes the registration inaccurate.

To improve the registration we increased the acceptance level of the inliers. Figure 15 and 16 shows two filtered images with different acceptance level of the inliers. Figure 15 show the image of what we used to get the results presented in Section 5 and Figure 16 is the 'optimal' amount of filtering but resulted in worse registration.

6.4 Difficult Environments

Other than the multipath interference and radial distortion complicating the registration, some environments proved harder to process than others. The multipath interference produced errors in the Door scene (Figure 6f) and similar scenes where many edges

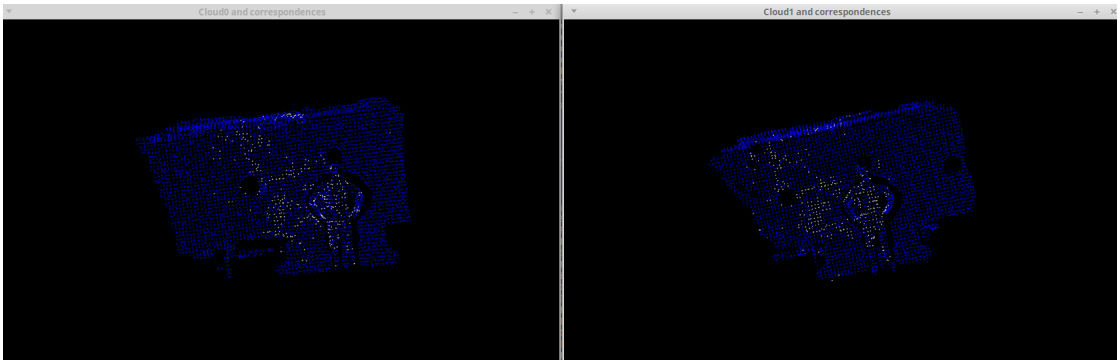

Figure 11: Two images of the same scene where the white points are the correspondences between the different images. In these images the noise is not removed.

Figure 12: The same point clouds as in Figure 11 from a different angle. Here it is clear that any of the correspondences are in fact the points which we define as outliers.

were present. Color can also be noted as an important factor where the camera had more trouble detecting points on darker surfaces as in the Furniture and Kitchen 1 scene (Figure 6c), that even though very similar to Kitchen 2 (Figure 6d) resulted in a lower amount of matches and higher RMS error as seen in Table 2.

Another important factor that can be assumed to have a large impact based on the results is the presence of objects. The Corridor scene (Figure 6a) that fulfills the parts of having bright surfaces and few corners and edges still gives low matches and higher error. This could be due to it not having the same amount of geometrical objects as other scenes. In the results of the Kitchen 2 scene, it can be noted that it has the highest number of matches and the lowest RMS out of all scenes in Table 2. This scene contains several objects, bright colors and few corners, that all are factors that are noted to affect the results.

For future work, some parameters on the camera could be more thoroughly tested to reduce some impact of these effects e.g. shutter time and light intensity.

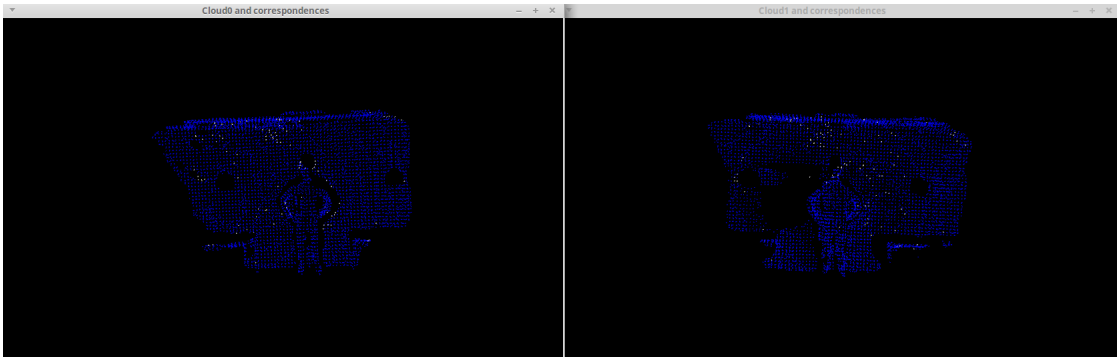

Figure 13: This is the same images as in Figure 11 but here the cloud has been filtered and all the outliers have been removed.

Figure 14: The same point clouds as in Figure 13 from a different angle.

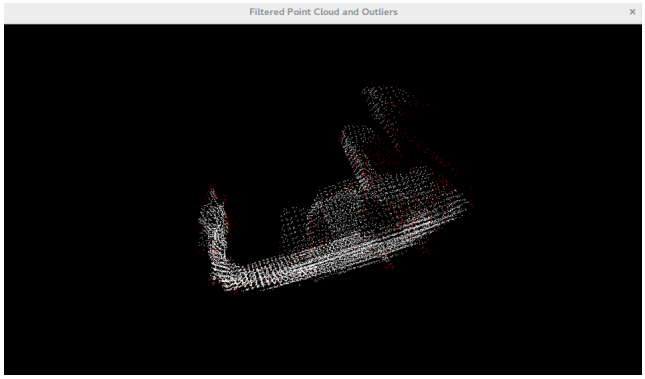

Figure 15: The amount of outliers removed to get the results presented in Section 5. The white points are the inliers and the red points represent the outliers.

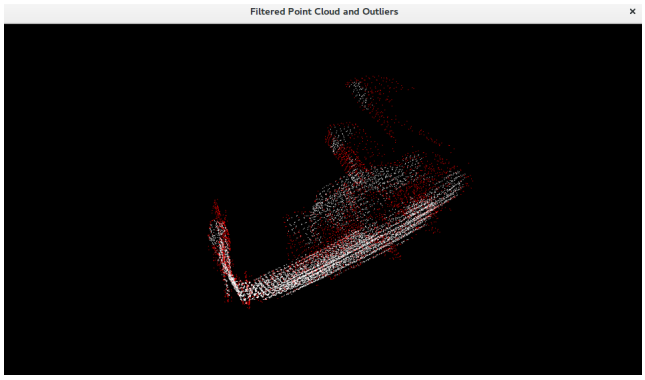

Figure 16: The amount of outliers removed in the case of 'optimal' outlier removal. This has not been used to produce the results in this report.

6.5 Overlapping Images

The experiments of using every second image in the data sets showed the same results for all scenes, that less overlap between images results in fewer correspondence matches and higher error. Fewer correspondences reduce computational times drastically but in the current state the qualitative results from using less overlap are not good enough. With further work the quality of registration could be increased by solving the problems of radial distortion and multipath interference in combination with the testing of parameters mentioned in 6.4. With this improved quality, less overlap can be used to increase speed.

7 Conclusion

The goal of this project has been to develop a program that can produce a 3D model of a room from a sequence of Time-of-Flight data. This goal has been achieved but the result varies depending on what scene is captured. By support from the discussion in Section 6 we have come to the conclusion that the amount of overlap between each image pair and the physical environment of the scene are two factors that affect the result of the registration. Regarding the overlap between the images in the sequence the conclusion is that the more overlap there is between each image pair, the better the registration. The physical environment is a somewhat more difficult factor. This includes corners, objects, materials, lighting and color variations between scenes. These physical factors will produce the artifacts and problems described in the discussion.

Furthermore, the overall quality of the registration could be improved by adding more pre-processing of the data acquired by the camera to reduce the impact of radial distortion and multipath interference. This leaves the field wide open for future work and further investigation.

References

[1] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European Conference on Computer Vision*, Springer, 2016, pp. 766–782.

[2] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp.," in *Robotics: Science and systems*, vol. 2, 2009, p. 435.

[3] M. Danelljan, G. Meneghetti, F. Shahbaz Khan, and M. Felsberg, "A probabilistic framework for color-based point set registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1818–1826.

[4] R. Rusu, N. Blodow, and M. Beetz., "Fast point feature histograms (fpfh) for 3d registration," *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[5] R. Rusu, N. Blodow, A. Holzbach, and M. Beetz, "Fast geometric point labeling using conditional random fields," *In Proceedings of the 22nd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[6] http://www.fotonic.com/faq/, "Fotonic faq," visited 2017-12-06.

[7] M. Munaro, F. Basso, and E. Menegatti, "Tracking people within groups with rgb-d data," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October,2012.

[8] E. Cappelletto, P. Zanuttigh, and G. M. Cortelazzo, "Handheld scanning with tof sensors and cameras," PhD thesis, Dept. of Information Engineering, University of Padova, Italy, 2012.

[9] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, Oct. 2009.

[10] D. Jiménez, D. Pizarro, M. Mazo, and S. Palazuelos, "Modeling and correction of multipath interference in time of flight cameras," *Image and Vision Computing*, vol. 32, no. 1, pp. 1–13, 2014.