A Performance Evaluation of a 3D MLI Time-of-Flight-sensor for applications in a domestic niche
(Bachelor Thesis)

Ruud Henken, s1634097, R.Henken@student.rug.nl,
Sjoerd de Jong*, Herman Kloosterman†

August 25, 2010

Abstract

When an autonomous robot is placed in an unknown environment, it does not yet have a map. To be able to build a map it is important that landmarks can be extracted from the environment. In this bachelor project we have attempted to extract landmarks from a domestic environment using a 3D MLI Time-of-Flight (ToF) sensor. The sensor uses the phase change of the reflection of actively emitted infrared light to determine the distance to objects. The sensor has a range from approximately 200 *mm to* 7000 *mm and a resolution of* 56 *by* 61 *pixels. We used the the sensor to build a dataset of gray-scale intensity images. We applied the SIFT feature detection method to detect and describe features in the gray-scale images and used SIFT matching to match the features. The experiment shows that by using this method it is quite difficult to extract reliable landmarks from the environment.*

1 Introduction

For autonomous robots it is important to have good performing sensors. Building a map of an unknown environment is an example where good performing sensors are needed. In order to build a map the robot needs to be able to extract landmarks from the environment. In this bachelor project we evaluate the performance of a 3D MLI Time-of-Flight (ToF) sensor on extracting landmarks from a domestic environment.

A Time-of-Flight (ToF) sensor is a sensor that can measures distance in a 3D surface by using light pulses. It actively emits infrared light pulses at a modulated frequency and measures the phase change of the reflected light. All the obstacles in the environment reflect the emitted infrared light and since the speed at which light travels is known, it is possible to measure the distance by using the phase change. With c being the speed of light, f the modulated frequency and ϕ the phase change, the distance d to the environmental obstacles is given by the formula [Feulner et al., 2009]: $d = \frac{c}{4\pi f}\phi$.

ToF-sensors are relatively new and yet they already have applications in 3D modeling and measuring distance to objects in a 3D surface. Related examples are 3D head tracking [Göktürk and Tomasi, 2004] and automated park assist [Gallo et al., 2008]. We are particularly interested in the overall performance of the ToF-sensor in domestic niche. Eventually we would like to be able to use the sensor for navigation and self localization for autonomous robots in a domestic environment. For this to be possible, we first need to be able to extract landmarks from the environment. More general we can say that in order to make an autonomous robot navigate in an environment it needs reliable input from the environment. Therefore good performing sensors are of great importance in the field of robotics.

*University of Groningen, Department of Artificial Intelligence
†PHILIPS, Drachten

So our goal is to find one feature detection method that performs well on extracting landmarks using solely a ToF-sensor. In order to find the best method we will first need to build a dataset. Setting up the dataset is an important part of this project and requires careful thought. We will discuss setting up the dataset in section 2.2. Once we have a dataset we can use it to evaluate the performance of several feature detection methods. To be able to make a comparison between several methods, we will use the following four evaluation properties:
a. Robustness, recognizability, repeatability
b. Uniqueness, quality match
c. Viewpoint invariance
d. Noise sensitivity
We will give a detailed explanation on these four properties in section 2.4.

Once we have build the dataset, we want to use feature detection methods to extract landmarks from the data. The feature detection method we will use is SIFT [Lowe, 2004]. We will apply the SIFT feature detection algorithm to detect and describe features and use SIFT matching to match the features. High expectations go to the SIFT algorithm, although the low resolution of the sensor could cause problems.

Since we want to make a straight judgment on the performance of the ToF-sensor, we think it is important to compare the results with other sensors too. Therefore an experiment with an IR-sensor is performed at the same time [Volger, 2010]. The results acquired by the IR-sensor are based on a dataset containing the same objects we use. The dataset contains recordings of several objects from different positions and/or angle. So in fact there are two datasets. One build with the ToF-sensor and one with the IR-sensor, but both containing data from the same environment. Throughout the whole project we aimed at comparing the results of both sensors. This means that the influence of willing to make a comparison is visible throughout the whole project. Although the aim was to make a comparison, we will not deal with it in this paper. We will discuss further details in Section 4.

In summary we are interested in the general performance of the ToF-sensor in domestic niche and would like to determine the best performing feature detection method for navigating an autonomous robot in domestic niche. Our research question therefore reads as follows. How well does SIFT together with the 3D MLI ToF-sensor perform on robustness, uniqueness, viewpoint invariance and noise sensitivity on extracting landmarks in a domestic environment? The hypothesis we make is that the 3D MLI ToF-sensor performs well enough on all four properties for an autonomous robot to be able to navigate trough the environment.

2 Methods

With this project we are working towards an autonomous robot that can navigate it's way trough every environment. The real challenge of course is to be able to operate in an unknown environment. This means that there is no predefined world model and there are no predefined landmarks. Without landmarks the autonomous robot is not able to navigate. This research is aimed at finding the best method for extracting landmarks from an unknown domestic environment using the 3D MLI ToF-sensor. Next we will discuss the sensor we used in section 2.1. Thereafter we will explain in detail how the dataset was set up in section 2.2 and the feature detection method we used in section 2.3. Then we will explain how to measure the performance in section 2.4 and explain the experiment in section 2.5.

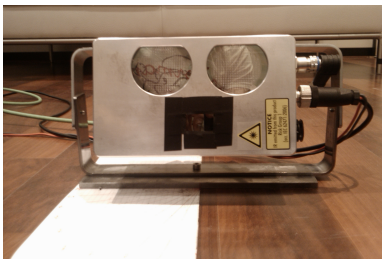

Figure 1: Philips's 3D MLI Time-of-Flight-sensor

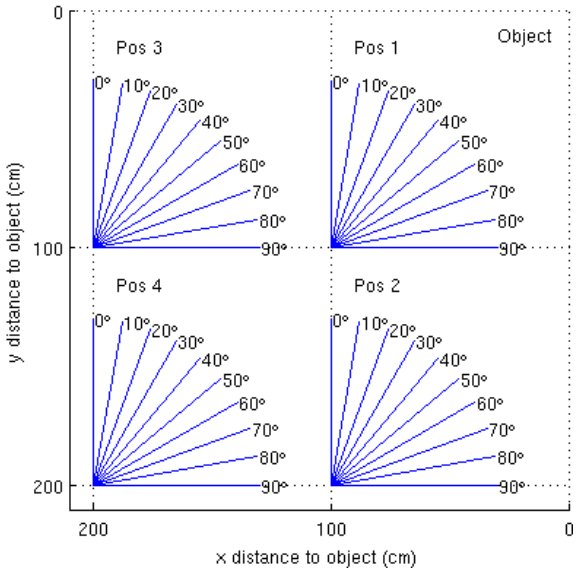

Figure 2: Schematic design of our dataset. Snapshots were taken from the four intersections (e.g., 100x100 cm, 100x200 cm, etc). With pos 1-4 being the number of the position we will use for reference.

2.1 Time-of-Flight-sensor

The sensor we used is a 3D MLI ToF-sensor from Philips (see Figure 1). The sensor has a range of approximately 200 mm to 7000 mm. Beyond these ranges the camera is no longer reliable. This has to do with the phase change of the actively emitted infrared light of the camera. The camera uses the phase change to determine the distance of an object. If an object is further away than 7000 mm the phase comes back at zero and you can no longer tell whether the object is close or far away. The same is true for measurements lower than 200 mm. For this reason we only took snapshots of objects within the range of 1000 mm to 3000 mm. The ToF-sensor can be used for recording video or taking single snapshots. We only use the snapshot functionality since it is more practical knowing the exact location at which a snapshot was taken. Every snapshot is a frame which is returned by the sensor and contains error and depth information per pixel. We converted the frame data to gray-scale intensity images. The gray-scale images are eventually stored in the dataset. More on converting the frames to images in section 2.2.1.

Figure 3: Gray-scale images from ToF-sensor recordings at 20, 30 and 40 degrees at position 1 of a small table.

A big disadvantage of working with ToF-sensors is the resolution. Most of the ToF-sensors still have a very low resolution. The 3D MLI sensor we used for this project has a resolution of 56 by 61 pixels. The low resolution might effect on the performance on the feature detection algorithm.

2.2 Dataset

To be able to perform this experiment we must first build a dataset. Building the dataset is an important part of this project. In fact, the first thing we did when we started it, was to make careful considerations on how the dataset should be build. Choices we made are based on a study by [Mikolajczyk and Schmid, 2005]. Their dataset includes image rotation and change of viewpoint, we will use a similar design of the dataset. We will look closely at change of position of the sensor and change of angle at which the sensor is aimed towards the objects. The objects we used are a bank, box, corner (of a room), pillow, plant, chair and a table.

An important requirement of the dataset is that it must be related to that of the IR-sensor in a way that makes it possible to compare the performance of the ToF-sensor with the IR-sensor. We have realized this in two ways. One is by taking recordings of the same objects, and the second is by recording objects from the same relative positions. For the latter we used a grid bases approach shown in Figure 2. An important difference between a recording of the ToF-sensor and the IR-sensor is the way the recording is taken. The IR-sensor can rotate a full 360 degrees on it is axis, while the ToF-sensor takes into account only one angle. In order to solve this issue we have manually rotated the ToF-sensor over a range of 90 degrees.

Table 1: Overview of recordings taken for each object

Object:	Position(s):
bank:	2, 3, 4
box:	1, 2, 3, 4
corner:	1, 2, 3
pillow:	1
plant:	1
chair:	2, 3, 4
table:	1, 2, 3, 4
total:	19

2.2.1 Generating gray-scale images

Table 1 lists all the recordings which are taken for the dataset. Each recording contains a total of 10 images (one for each angle, 0 to 90 degrees), therefore the dataset consists of a total 190 gray-scale intensity images. The images are build from the frame data which comes directly from the ToF-sensor. Each frame the sensor returns, contains per pixel error and distance information. If the measurement of a single pixels goes beyond the scope of the sensor the error value will be unequal to zero. The distance values returned by the sensor always lie between 0 mm and 7000 mm. To build a gray-scale image we look at every flawless pixel in the returned frame to determine the minimum and maximum distance value. The minimum and maximum values are then used to scale every distance value to a value between 0 and 1. These scaled values are then multiplied by 255 to create a gray-scale intensity value. And finally the gray-scale intensity value is assigned to the pixels in an intensity images with observations far away being white and observations close being black. If a pixels happens to have an error, we assign it a white value for being far away. Images will eventually look like Figure 3. The most left image is taken from an angle of 20 degrees, the middle image from an angle of 30 degrees and the right image of an angle of 40 degrees.

Since we do not have recordings with added noise and yet we do want to test the influence of noise, we manually added noise. Noise is added by adding a random number between -250 mm and 250 mm to the distance value.

2.3 Feature detection

At this point we have a dataset with 190 gray-scale intensity images. The images can then be used with feature detection methods to extract points of interest. The feature detection method we will use is Scale-invariant feature transform (SIFT) [Lowe, 2004]. With SIFT the original images is convolved with Gaussian filters at different scales to blur the original image. The blurred image is then compared to the original image (Difference of Gaussian) to detect the interest points, which are called keypoints in SIFT. The maxima/minima of the Difference of Gaussian are the most interesting points and are the points where the keypoints are taken. A SIFT description of the keypoint consists of an description of the region surrounding the keypoint. Eventually the description of the region is stored in a vector of length 128. The vector is a description of the interest point and can than be used to match it with others. A match exists if the Euclidean distance between two descriptor vectors is lower than a specific threshold t. The optimal value of t is found by running a few experiments. More on finding the optimal value of t in section 2.5.

When SIFT is applied on normal images, keypoints are usally found on rapid transitions in the image like the end of a black line on a white wall. Our dataset consists of gray-scale intensity images. Hence, in our dataset the rapid transitions occur when there is a rapid transition in depth. So for example it would be likely that keypoints are found on the edge of an object where depth changes quickly.

2.4 Performance Evaluation

In this section we will give four evaluation criteria that give us the possibility to compare several feature detection methods and make it possible to compare the results of the ToF-sensor with the results of the IR-sensor. Although we have only tested one feature detection method we will now define a standardized method to judge the performance of the feature detection method. The following four evaluation criteria give us the ability to evaluate the performance of the feature detection method and compare it to others:

a. Robustness, recognizability, repeatability
Robustness says something about how 'strong' a feature actually is. When the viewpoint of the sensor is slightly changed or the distance between the sensor and the object is changes the feature should still be visible. A feature is said to be robust if it can be seen from more than just a single viewpoint. An experiment on determining the recognition rate of each object is performed to find the robustness.

b. Uniqueness, match quality
When an object of the same type is presented, it is expected to give a match (e.g., an image of a table should give a match with another image of the table taken from the same viewpoint). If a different object is presented we expect it not to match. A descriptor is said to be unique if it only matches with descriptors it is supposed to match with. Hence, a descriptor obtained from an image of a table should not match with a descriptor obtained from an image of a plant. We will perform an experiment on finding the best match for each image to determine the uniqueness.

c. Viewpoint invariance
When navigating trough the environment, landmarks will probably be detected from different viewpoints. For optimal performance it is important that a landmark can still be detected if the viewpoint is changed. Hence, we want to know how well it performs on matching objects seen from a different angel and/or position. We will use the same experiment used in (a) for determining the viewpoint (in)variance.

d. Noise sensitivity
Here we want to test the influence of noise on the performance of the feature detection method. We want to know whether noise influences the performance. Since no noisy recordings were taken, we created noise by adding a random number between -250 and 250 to the per pixel distance value returned by the sensor. To find the noise sensitivity of the feature detection method we perform the same experiment as in (b), but with noise added to all images in our dataset.

2.5 Experiment

In total we will run two experiments to answer the four evaluation criteria stated above. First we will discuss the 'best match'-experiment used for determining the performance of (b) and (d) and thereafter we will discuss the 'recognition rate'-experiment used for determining the performance of (a) and (c).

2.5.1 Best match (b and d)

A nice way to evaluate the performance is by finding the 'best match'. Two descriptors match if the Euclidean distance between the descriptor vectors falls below a certain threshold t. The 'best match' is then defined by the match with lowest Euclidean distance between two descriptor vectors. The best match is found by comparing a descriptor vector found in one image to all other descriptor vectors found in the rest of the dataset. The matching descriptor vector with the lowest Euclidean distance is selected as best match. The optimal value for t is found by performing the experiment a couple of times and vary t towards an optimum.

A 'set' contains all the images of the positions from a particular object (e.g., all the images from the table from all four positions). A false match or mismatch is then defined as an image matching with another image outside of its own set (e.g., an image of a table matching with one of a plant). A correct match is then defined as a match between two images of the same set (e.g., an image of a table matching with an image of a table).

Each object can then be given a match score. With $\#CM$ being the number of correct matches and $\#FM$ being the number of false matches, the match score is defined as follows:

$$match\ score\ =\ \frac{\#CM}{\#CM\ +\ \#FM} \qquad (1)$$

We will use the match score to judge the performance of both (b) uniqueness and (d) noise sensitivity.

(b) Uniqueness can be evaluated from this experiment since a descriptor is said to be unique

if it only matches with images of the same set. In other words, the match score is a measure for uniqueness of a descriptor.

(d) noise sensitivity can be evaluated from this experiment by performing the same experiment once more, but with noise added to all of the images in our dataset. If noise influences the results in a negative way, it is expected that the number of times a correct match is made will decrease. Hence, if we perform the experiment once more, but with noise added, we can determine the influence of noise on the performance of the feature detection method.

In determining the best match we (of course) let out matching an image with itself. It is no surprise that images match with themselves and evidently create a plot with a perfect diagonal. The only reason why it can be interesting to match images with themselves is to gather information about images in our dataset that do not contain any descriptors at all.

2.5.2 Recognition rate (a and c)

Another experiment is performed to find the recognition rate of each object. The recognition rate is calculated for each position of each object (the dataset does not contain enough data to calculate the recognition rate for the two objects plant and pillow). The result is set out in a graph with on the x-axis the position of the sensor and on the y-axis the recognition rate. If N is the number of positions from where a recording of an object is taken and I the number of correctly recognized objects, the recognition rate RR for an object Obj is defined by [Ypma, 2007]:

$$RR_{Obj} = \frac{1}{10} \sum_{\theta=0}^{(N-1)*10} I \qquad (2)$$

The recognition rate gives information on both the (a) robustness of the descriptors and the (c) viewpoint invariance. A descriptor is said to be robust if it can be recognized from more than a single position. Hence, if the plot for the recognition rate shows a flat horizontal line, it means an object is recognized from all the positions an equally amount of times.

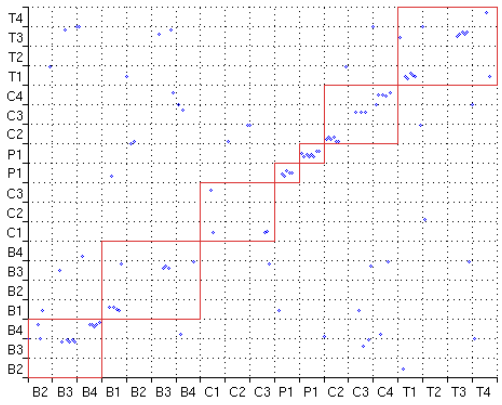

Figure 4: Best match results matching all images

The recognition rate gives a direct measurement for (c) viewpoint variance. If the viewpoint changes while the feature detection method is insensitive for viewpoint, there should be no change in recognition rate.

3 Results

In this section we will discuss the results on all four of the evaluation criteria. We have conducted two different experiments, but we will discuss them by the four evalutation criteria.

a. Robustness, recognizability, repeatability
Figure 5 gives the resulting recognition rate for all the objects. The Figure contains five plots, each for one object. In total we used seven objects in our dataset, but for two of them we do not have enough data to calculate the recognition rate. The x-axis in each plot represents the position at which a recording is taken (position 1-4, see Figure 2). The y-axis gives the recognition rate for the particular object at the particular location according to Equation 2. A feature is said to be robust if it can be seen from more than just a single viewpoint. If an object is robust, we would expect the plot would give a flat horizontal line since an object is recognized equally well from all positions. In our case none of the lines are flat.

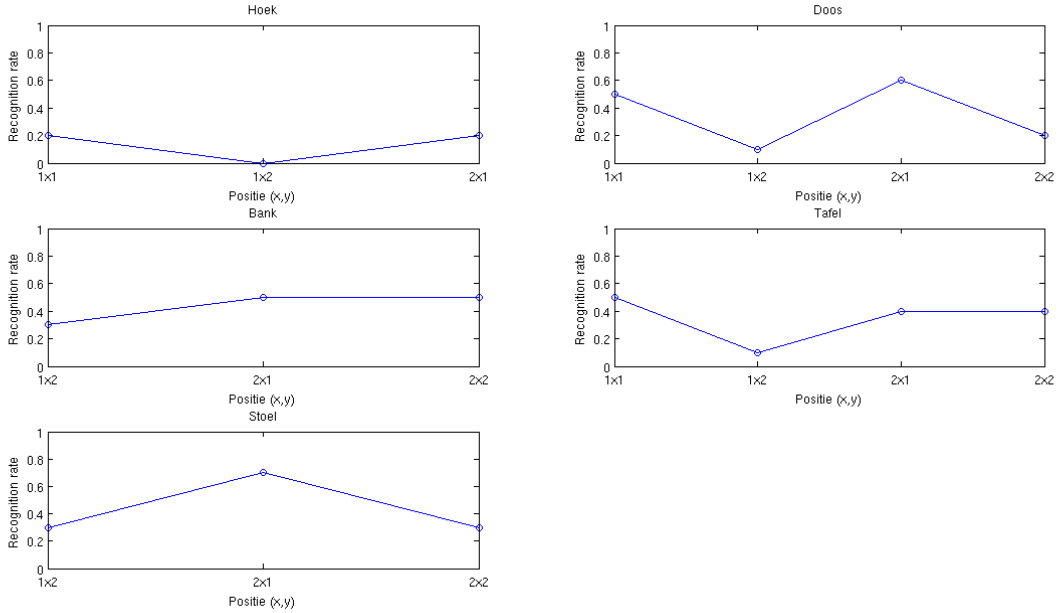

Figure 5: Recognition rate against sensor position for corner, bank, chair, box and table.

This would suggest that the obtained features are not robust.

b. Uniqueness, match quality

The results of the 'best match'-experiment are shown in Figure 4. Each column and each row contains information about an object and a position (e.g., the fourth column contains information about the box at position 100 x 100 cm). Also, each column and each row contains information about a total of ten images, ranging from 0 degrees to 90 degrees. So each column and each row can contain up to a maximum of ten blue dots. The plot also contains seven red squares, one for each set of objects. In the most ideal situation we would like to see that every blue dot is close to the diagonal or at least within the square for the set it belongs to.

As we can see in Figure 4 there are quite some dots missing and there are also quite some mismatches. For example, the third column of the box matches twice with images taken from the table and the images from the box hardly match with any other images at all. The reason that not every column has the same number of matches has to do with the threshold t and the descriptors obtained from the images. If the difference between two descriptors does not drop below the value of t, no match takes place and hence there is no best match. The same is true for images with only a few or even no descriptors. If no descriptors can be obtained from an image, no matching can take place.

c. Viewpoint invariance

For measuring the viewpoint invariance we use Figure 5, the same as for measuring (a) robustness. It is expected that if an object is invariant for viewpoint variations, the plot would give a flat horizontal line. We would expect it to be like this since if it is invariant to viewpoint changes, the recognition rate does not change when viewpoint is changed. Like we already concluded in (a), none of the lines are flat. This would mean that every position of an object has a different recognition rate and the method suffers from viewpoint variations.

d. Noise sensitivity

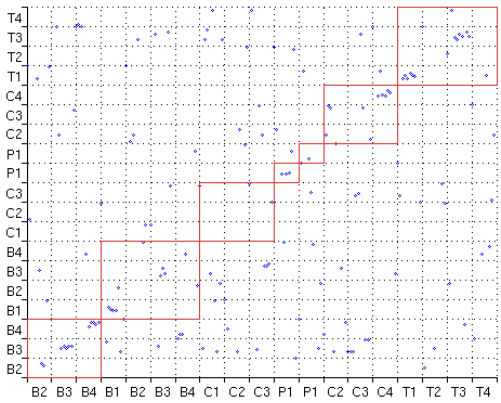

Figure 6: Best match results matching all images with artificially added noise

Table 2: Matching scores

	Match score	
Object:	Normal:	Noise:
bank:	0.43	0.43
box:	0.23	0.28
corner:	0.13	0.07
pillow:	0.50	0.30
plant:	0.80	0.10
chair:	0.47	0.33
table:	0.35	0.44
overall:	0.42	0.27

Figure 6 shows the results for best match after adding noise. It is clearly visible that in contrast to the results shown in Figure 4 more mismatches are made when noise is added. Table 2 summarizes the matching results of normal matching conditions and the condition where noise is added. The most left column contains all the objects in our dataset. The middle column shows the matching scores for every object under normal conditions. The right column shows the matching scores for all the objects while noise was added. It is clearly visible that the results greatly decrease when noise is added. The overall matching score for objects under normal conditions is 0.42 and the matching score with added noise is 0.27. The only objects where the score is slightly improved when noise is added are the box and table.

In the next section we will discuss the remarks on this study, draw a preliminary conclusion and we will give a suggestion for further research.

4 Discussion

4.1 Remarks

4.1.1 Dataset

A few remarks must be made at this study. First of all there is the dataset. As of the beginning of this study we were aware of the importance of careful thought for setting up the dataset. The better the dataset was set up, the easier it would be to perform an experiment and draw meaningful conclusions. Maybe the most important shortcoming of our dataset is that it is incomplete. We do not have recordings of every position for all the objects and this makes it impossible to fully amplify some of the evaluation criteria. For example we can not compare the influence of moving the sensor from a 100 cm to a 200 cm to moving the sensor from 200 cm to a 300 cm. Another serious error is how the recordings were made. All of the recordings were taken from objects that were placed in the very same corner of the test room. This could cause all of our results to be invalid, since it might not be the actual objects we were matching, instead it could have been environmental keypoints that caused the matches or even mean the distance tot the object. This is a serious problem and at this point we can not exclude that this has happened. However if we take a quick look at where the descriptors are located, we can see that this is not likely to have corrupted the results. It must be said that half of the time a mirror was visible in the corner of the test room. A mirror could greatly influence the results since the infrared-light is not reflected back at the sensor as it does with other objects, but instead it is reflected to infinity (or at least beyond the range of the sensor).

4.1.2 Feature detection methods

Secondly we have only tested one feature detection method on extracting landmarks from ToF-data. Maybe SIFT does not work very well with this

kind of data and therefore does not show very good results. We wanted to compare the results of several feature detection methods so we would be able to compare their differences and select the best performing method. Not only did we want to compare several feature detection methods, we also wanted to compare the results to those of the IR-sensor. At the same time an experiment was done with an IR-sensor with triangulation to extract landmarks [Volger, 2010]. The idea was to compare the results of that study with the results this one. The way we set up the dataset eventually made it impossible to compare the results.

4.1.3 Practical concerns

And finally there are some practical concerns. The sensor we used is quite heavy to carry around for a autonomous robot and could therefore cause some problems. The sensor also uses a lot of energy since it needs to be connected to the wall plug all the time. When active it requires 230 V. Another thing is that it is a relatively expensive sensor to put in an autonomous robot. One final thing to mention is that it is discouraged to look into the infrared lights of the sensor since it could cause headaches. The emitted light is not visible to the human eye, but still it could cause headaches. This fact is rather impractical since the sensor will need to be used in domestic environment and people will most likely look at it when it is operating.

4.2 Conclusions

a. Robustness, recognizability, repeatability
The experiment on finding the recognition rate shows that by changing the viewpoint of the sensor, also influences the recognition rate for each position. With robustness meaning keypoints are visible from more than a single viewpoint, this suggests that extracted keypoints are not robust and therefore the feature detection method scores poorly on robustness.

b. Uniqueness, quality match
A keypoint is said to be unique if it only matches with keypoints from the set it belongs to. In terms of the experiment it would mean that matches are only allowed within its (red) square. The experiment shows that quite some mismatches are made. Some objects perform better the other (e.g., on normal conditions the plant has the highest match score and the corner the lowest match score). But for all objects it seems no perfect match can be made, therefore we can only say that it is hard to obtain unique keypoints using ToF-data.

c. Viewpoint invariance
Here the same is true as for robustness. The experiment on finding the recognition rate showed that viewpoint change effects the performance of the method and therefore scores poorly on viewpoint invariance.

d. Noise sensitivity
Noise was added manually to determine the performance of the sensor with noise added. We used the best match experiment to determine the influence of noise. The experiment showed that noise greatly influenced the best match results. Performance on most of the objects decreases dramatically, the only object where the performance slightly improved were the box and the table.

The conclusions that we can draw from this study is that a 3D MLI ToF-sensor together with SIFT description and SIFT matching in our implementation does not give reliable enough landmarks. The method scores poorly on the four evaluation criteria we have postulated in Section 2.4. This conclusion corresponds with earlier research by [Ellekilde et al., 2007]. They have found that using only a CSEM SwissRanger SR-2 it is not possible to extract reliable landmarks using the SIFT algorithm due to its low resolution.

4.3 Further Research

Our suggestions is that further research should be aimed at extracting 3D landmarks instead of extracting landmarks from an intensity images. If 3D landmarks can be obtained it would give a more unique descriptor and would therefore have more unique matches. The problem with our method is that by changing the viewpoint the intensity images changes as well. By creating 3D description of a landmark we think it should be more stable in matching since the relationship between keypoint

distances stays the same. Even if you change viewpoint. So our suggestion for further research is to look after extracting 3D landmarks.

References

[Ellekilde et al., 2007] Ellekilde, L.-P., Huang, S., Miro, J. V., and Dissanayake, G. (2007). Dense 3d map construction for indoor search and rescue. *Journal of field robotics*, 24:71–89.

[Feulner et al., 2009] Feulner, J., Penne, J., Kollorz, E., and Hornegger, J. (2009). Robust real-time 3d modeling of static scenes using solely a time-of-flight sensor. *IEEE-Computer-Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 340–347.

[Gallo et al., 2008] Gallo, O., Manduchi, R., and Rafii, A. (2008). Robust curb and ramp detection for safe parking using the canesta tof camera. *IEEE*.

[Göktürk and Tomasi, 2004] Göktürk, S. B. and Tomasi, C. (2004). 3d head tracking based on recognition and interpolation using a time-of-flight depth sensor. *Conference on Computer Vision and Pattern Recognition*, pages 211–217.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[Mikolajczyk and Schmid, 2005] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 27(10):16.

[Volger, 2010] Volger, M. (2010). Environmental feature extraction and comparison using a rotating infra-red sensor.

[Ypma, 2007] Ypma, J. (2007). Improving sift for 3d object recognition using active vision and clustering. Master's thesis, Rijksuniversiteit Groningen.