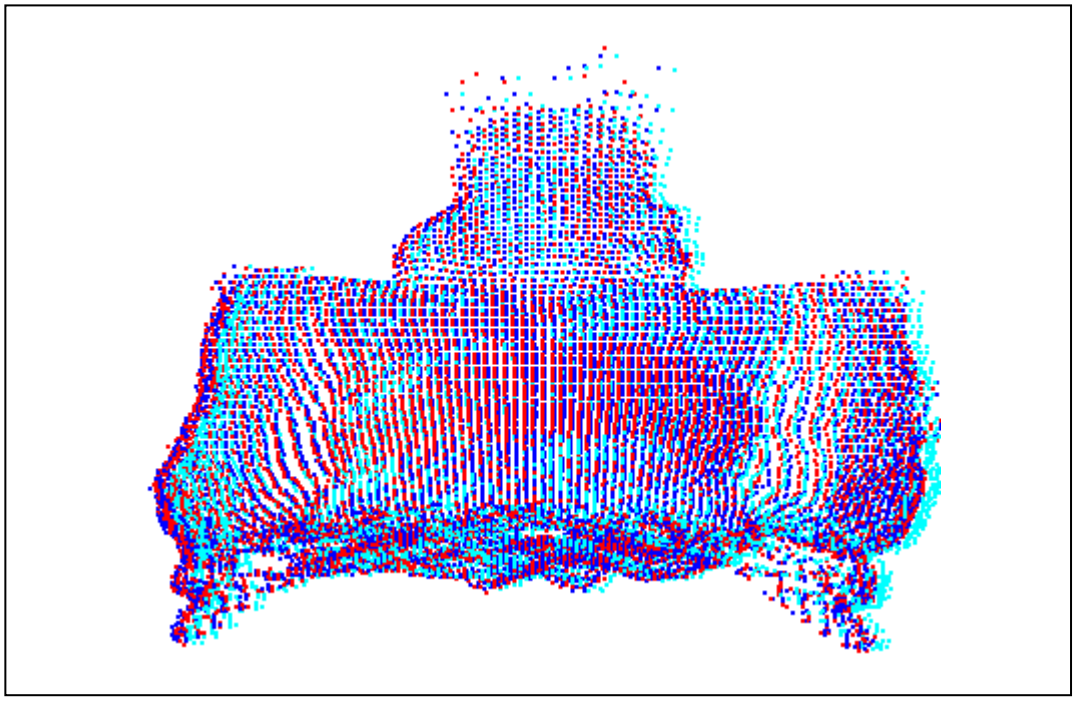
CHALMERS



Evaluation of 3D camera technology for patient positioning within the field of radiation therapy

Master of Science Thesis

ARMIN BÖLLER

Department of Computer Science and Engineering
Division of *Computing Science*
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2010

Evaluation of 3D camera technology for patient positioning within the field of radiation therapy

by

Armin Böller

A thesis submitted to

Chalmers University of Technology

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science and Engineering

Chalmers University of Technology

Gothenburg, August 2010

Evaluation of 3D camera technology for patient positioning within the field of radiation therapy

by Armin Böller

Chalmers University of Technology
Department of Computer Science and Engineering

ABSTRACT

The thesis describes shortly the Time-of-Flight technology and also the experiments, algorithms and results for the registration of point clouds acquired by a Time-of-Flight camera. The aim is to explore possible applications of the ToF camera for patient positioning in the field of radiotherapy. A distance function between two point clouds has been defined for the registration of point clouds in different frames. Afterwards the distance of the reference frame with a linear transformed point cloud is minimized. For this purpose the UOBYQA (unconstrained optimization by quadratic approximation) algorithm has been used. The following results could be obtained: a single pixel of the camera itself had a standard deviation of 0.5mm; the translation of a spherical object could be determined with an average error of 0.5mm. The algorithm for the registration of arbitrary surfaces with six degrees of freedom are not completely finished and will be improved in further work. The cameras are sensitive to temperature fluctuations, which has been tried to be avoided by using two cameras at the same time. For this purpose the cameras have been calibrated and their coordinate system transformed from one to the other. Finally a motion surveillance application is presented, which creates a real time video stream with a visualization of the distance from the actual frame to the reference frame.

ACKNOWLEDGMENTS

Ich möchte mich bei Professor Peter Damaschke für das betreuen der Masterarbeit bedanken. Jag ville gärna studera i Sverige och Chalmers tekniska högskola blev då ett naturligt val för mig. Ich möchte mich bei Stephan Erbel und Doktor Kajetan Berlinger für die Betreuung bei Brainlab bedanken. I would like to thank Brainlab for giving the possibility of writing my master thesis. Ich möchte mich bei meiner Familie für die Unterstützung bedanken. And finally I would like to thank all my friends for having such a nice time, while I was writing my thesis.

Contents

Table of Contents **iv**

1 Introduction **1**
1.1 State of the Art . . . 1
1.2 Time-of-Flight Camera . . . 3
1.3 Application of the ToF Camera . . . 4
1.4 Recent Work . . . 4

2 Time-of-Flight Technology **5**
2.1 SwissRanger4000 . . . 6
2.2 Error Sources . . . 6

3 Sphere Matching **7**
3.1 Experiment Setup . . . 7
3.1.1 Phantoms . . . 8
3.1.2 Tested Data . . . 8
3.2 Calculation of the Sphere Center . . . 9
3.2.1 Hough Circle Transform . . . 9
3.2.2 Local Minima Search . . . 10
3.2.3 Sphere Fitting . . . 10
3.2.4 Candidate Filtering . . . 12
3.2.5 Iterations to the Final Coordinates . . . 12
3.3 Result . . . 15
3.3.1 Static Observation . . . 15
3.3.2 Shifted Sphere . . . 16
3.3.3 Temperature Influence . . . 18
3.4 Discussion . . . 18

4 Stereo Camera System **20**
4.1 Experiment . . . 20
4.2 Transformation Matrix . . . 21
4.3 Algorithms . . . 21
4.3.1 Calibration . . . 22

4.3.2 Combining the Positions 22
4.4 Result 22
4.5 Discussion 23

5 Surface Matching **26**
5.1 Experiment 26
5.2 Optimization Algorithm 27
5.2.1 Point to Point Cloud Distance 27
5.2.2 Objective Function 28
5.2.3 Approximation Algorithm 29
5.2.4 Starting Value 30
5.3 Result 30
5.4 Discussion 32

6 Patient Surveillance **35**

7 Conclusion **37**

Bibliography **38**

List of Abbreviations

3D	Three Dimensional
CT	Computer Tomography
DRR	Digitally Reconstructed Radiograph
IGRT	Image Guided Radiation Therapy
LED	Light Emitting Diode
LINAC	Linear Accelerator
PMD	Photonic Mixer Device
UOBYQA	Unconstrained Optimization By Quadratic Approximation
ToF	Time-of-Flight
RGB	Red Green Blue color level

Chapter 1

Introduction

Radiation therapy is one of the major treatment methods of cancer sufferers, besides surgery and chemotherapy. In the therapy are several sessions of radiation treatments used to destroy the tumor. At first a computer tomography (CT) scan is done to determine the affected tissue. This is followed by the treatment with the linear accelerator (LINAC). The radiation is emitted from different directions with the focus in the tumor to avoid the destruction of healthy tissue. To ensure that all tumor cells are destroyed and cannot regrow after the therapy, a larger volume than the tumor itself needs to be irradiated. The tumor could have grown since the last CT-scan or might not have been fairly visible at the time of the CT-scan. Further errors like the positioning of the patient relative to the treatment machine extend the required safety margin even more. In radiation therapy the time in-between the treatment sessions gives the healthy tissue some time to regenerate and the repeated positioning of the patient partially averages out the errors.

There are several techniques to minimize the positioning error. One possibility is to fix the patient with a frame or another one is to position and monitor the patient with help of image processing, a so called Image Guided Radiation Therapy (IGRT). Together with a precise LINAC and treatment plan, it is sometimes possible to reduce the number of treatment sessions to a single one. In this case a radiologist speaks about radiation surgery instead of radiation therapy. This speeds up the healing process, reduces the emitted radiation and may reduce the side effects of the therapy [1].

1.1 State of the Art

One commonly used IGRT system is ExacTrac® of Brainlab, which is part of the widely used Novalis Tx system (figure 1.1) [2]. Reflective Markers are placed onto the patient and tracked by a stereo infrared camera system. After the tumor is placed roughly in the focus of the LINAC, two x-ray acquisitions are done from different sides. These images are matched with digitally reconstructed radiograph (DRR) virtual X-



Figure 1.1 Brainlab ExacTrac® system with both X-ray tubes activated.

ray images generated from the planning CT-data, which includes the position of the tumor. The patient is automatically moved according to the distance vector between the LINAC focus and the tumor. Another X-ray acquisition is done to ensure a high precision with a maximum alignment error of 0.5 mm before the radiation treatment starts. For a tumor in a moving body part, as for instance the lung, a so-called gating can be used to reduce the amount of irradiated healthy tissue. During the gating the LINAC emits radiation while the tumor is in the focus. When the tumor is moving out of focus the treatment pauses and resumes as soon as the tumor is back in focus. To observe the movement, a camera system is needed. In the Brainlab ExacTrac® system 5 to 8 reflective markers are fixed to the patient's skin. The clinical personal needs to take care, that the markers are attached to body parts - moving with the tumor and are visible for the camera system. A benefit would be a camera system, which works without markers. This would speed up the patient setup and would avoid mistakes by placing the markers on wrong positions. An IGRT system without markers is AlignRT® from VisionRT [3] [4]. It uses two stereo camera systems, which reconstruct the surface of the patient to be able to place the patient in the same position as in the last session. There is no matching of the CT-image with the inner body parts through an X-ray scan. So this system is especially suited for a cancer, which is close to the surface, for example breast cancer.



Figure 1.2 Mesa SwissRanger4000 ToF camera mounted on a ladder.

1.2 Time-of-Flight Camera

An alternative sensor for an IGRT system is the Time-of Flight (ToF) camera technology (figure 1.2), which has been developed over the last years. The ToF camera system uses a light impulse to determine a three dimensional point cloud with distance information for each pixel. This can be done in real time in a video frame rate. The ToF camera system uses an array of light emitting diodes (LED) to send out near infrared light impulse. The light gets reflected at the objects in the field of view and returns partly to the ToF sensor. There are two possible ways to determine the distance to the object for each pixel. The possibilities are either by measuring the travel time, the light needed from the LED back to the ToF sensor, or the PMD (Photonic Mixer Device) principle with help of the phase-shift of the light. Depending on the distance from the LED to the reflecting object and back to the sensor the light phase has changed. By measuring the shift, the distance can be determined (Figure 1.3). Most of the ToF systems use the phase-shift measurement to avoid the use of a high precision clock. But depending on the modulation frequency there is a maximum unambiguous distance range. E.g. for the MESA SwissRanger4000 with a modulation frequency of 30MHz the maximum distance is 5 meter. With a modulation frequency of 15Mhz the unambiguous distance is 10 meter [5].

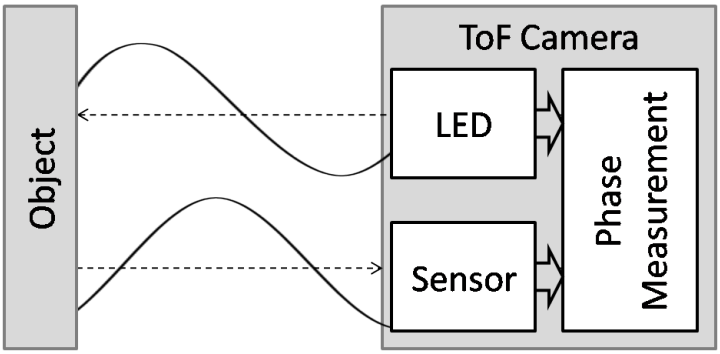


Figure 1.3 PMD measurement principle.

1.3 Application of the ToF Camera

The ToF point cloud of a radiation patient can be matched with the surface part of a CT-image. Together with the information of the inner body parts from the CT-image, the position of the tumor can be estimated. The body movements can be determined in real time and the distance information makes the image segmentation easy and robust. Target of this thesis is to determine how precise a point cloud obtained by a ToF camera can be registered with a surface model.

1.4 Recent Work

Similar work has recently been done by Schaller [6], who observed the breathing curve of a patient. Studies about the error rate of the raw signal were done in [7]. There are also studies to improve the result of the ToF sensor, either by further calibration [8], or by a superresolution method [9] or by combining the ToF sensor with a stereo camera [10].

The ToF sensor in general has a wide field of image analysis applications, i.e. space docking [11], surveillance systems [12], robotics [13] or in the automotive industry [14]. Microsoft is developing a game console with help of a CMOS distance sensor from PrimeSense [15], which does not work with pulsed light as the PMD sensor does and is much cheaper.

Chapter 2

Time-of-Flight Technology

More details about the Time-of-Flight technology are provided in this chapter. The principle of the ToF sensor functionality is described in section 1.2. The benefits of a ToF camera are the distance information for each pixel and consequently the easy and robust segmentation of different objects together with a high frame rate. As part of this thesis, the CamCube 2.0 from PMD Technologies and the SwissRanger4000 from Mesa Imaging have been tested. Both ToF cameras are measuring the phase shift of the infrared light with a PMD sensor. Due to a non disclosure agreement with PMD Technologies only results from Mesa's SwissRanger4000 are published in this thesis.

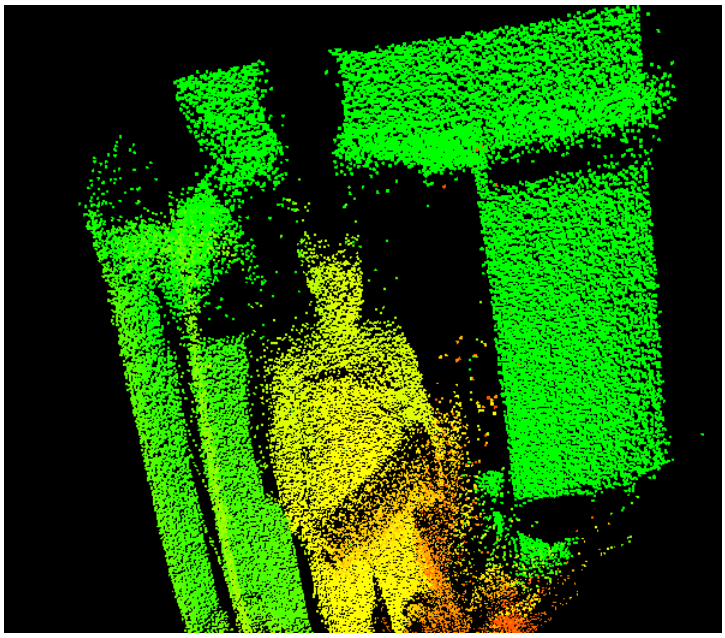


Figure 2.1 A ToF acquired point cloud of a person with a keyboard in the hand, staying in front of a wall.

2.1 SwissRanger4000

The SwissRanger4000 has a resolution of 176 x 144 pixels over 43.6° x 35.6° field of view, with a maximum of 54 fps and a maximum distance of 10 meter. To illustrate the distance of each pixel in the obtained point cloud, colors can be used. In the frame of figure 2.1 red to orange is in close distance to the camera, yellow in a medium distance and green in a higher distance.

2.2 Error Sources

To receive a valuable 3D point cloud from the camera there are several conditions to fulfill. For a high precision measurement it is important that a large amount of light is reflected back to the sensor. Therefore, black surfaces are leading to unreliable results, since these absorb a large amount of light. A higher integration time allows the sensors to receive a larger amount of light but on the contrary an oversaturated pixel sensor does not provide any valuable information at all. Dark objects might return enough light to the sensor with a high integration time, but close white objects may cause oversaturated pixels. Infrared light from the LED which gets reflected over a multipath (figure 2.2) falsifies the result as well [5]. Due to the complex technology every pixel is affected by a high noise ratio, but an averaging over several pixels and/or frames can increase the accuracy significantly. Furthermore sunlight contains a large amount of light of the same wavelength as used by the ToF technology and yield in further inaccuracy, but artificial light does not provoke this problem [16].

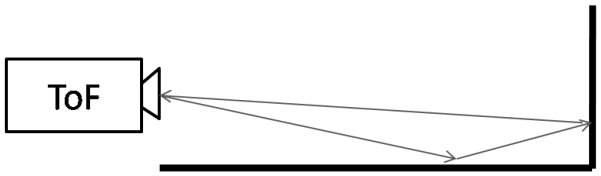


Figure 2.2 Multipath leads to false measurements.

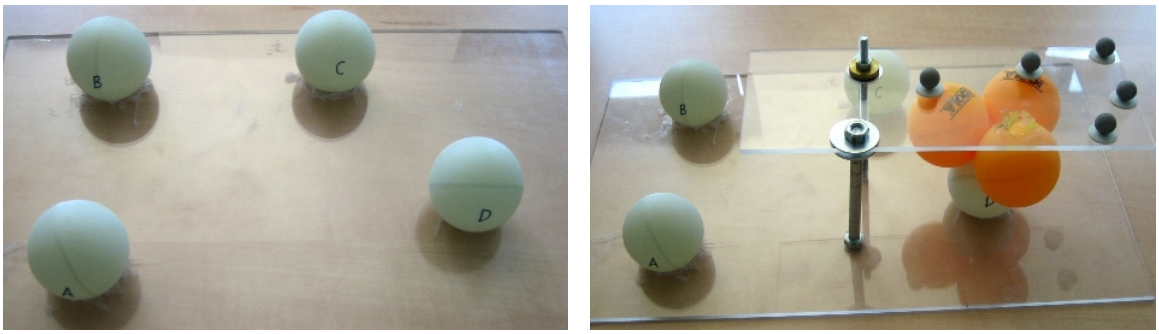
Chapter 3

Sphere Matching

Before real patients getting observed by a ToF camera, tests are required to determine how exact the camera works in an ideal environment.

3.1 Experiment Setup

A single acquired pixel itself is noisy, so a better result is achievable when using several pixels. If a geometrical model of the acquired object is fitted into a point cloud, the error can be averaged over all used pixels and reduce the noise ratio. A sphere is geometrical easy to describe and looks identical from every angle and is therefore a good object to test with. Afterwards, the obtained point cloud is matched with the geometrical sphere model to receive the center of the sphere. This is done several times to determine the accuracy of the camera.



(a) Phantom (b) Phantom with measurement device on top

Figure 3.1 Self constructed phantom out of four table tennis balls.

3.1.1 Phantoms

Four extra large table tennis balls with a diameter of 55mm are glued to an acrylic glass plane (figure 3.1a). To measure the distance between the four spheres, the camera system of Exactrac® was used with help of a self constructed measuring tool (figure 3.1b) with known geometry. The device was used to measure the distances between the white spheres with only a small uncertainty. For the testing, three of the spheres were used to create an orthogonal coordinate system and the position of the fourth sphere in this coordinate system can be measured.

The table tennis balls are quite small and in one meter distance to the camera, they return only around 80 pixels. To get a better measurement and to use a phantom which has a similar size as a human head another phantom has been used, a large styrofoam sphere (figure 3.2).

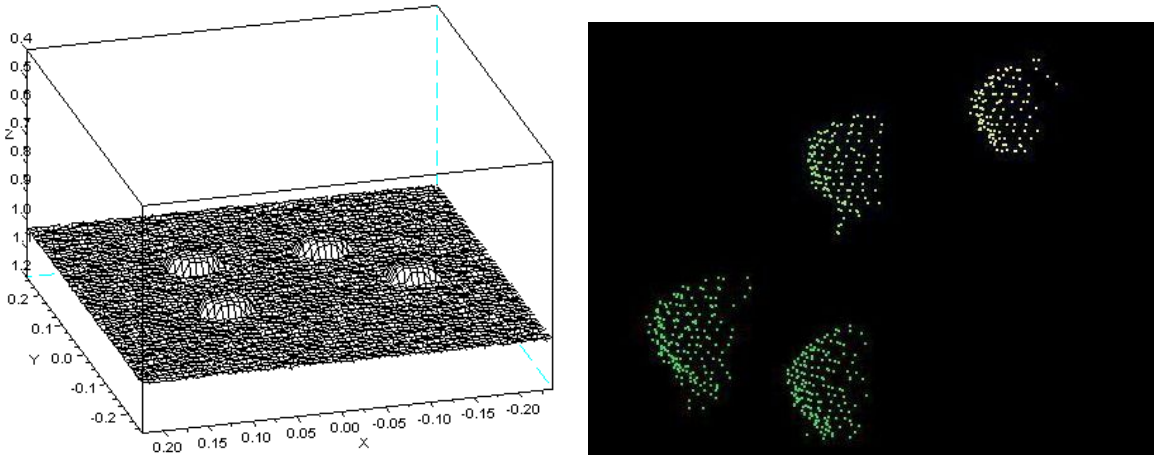


Figure 3.2 Styrofoam sphere to measure with a diameter of 250mm.

3.1.2 Tested Data

To be able to test the software before the ToF camera is available, an artificial point cloud has been used (figure 3.3a). The point cloud simulates the view of the camera from directly above the table tennis ball phantom (section 3.1.1) on a plate.

A point cloud acquired by the ToF camera looks like in figure 3.3b. Pixels, which values were determined by a signal of low amplitude, are filtered out, as well as pixels in a larger distance than 1.2 meters to save computation time.



(a) Artificial sphere image with added gaussian noise

(b) ToF point cloud of the phantom

Figure 3.3 Visualized input data for the sphere matching algorithm.

3.2 Calculation of the Sphere Center

To determine the center points of the spheres, a sequence of calculations is necessary. The first task is to detect, which pixels of the point cloud are part of the sphere surface. To do so, a sphere center is estimated with the local minima search (section 3.2.2). Afterwards all pixels, which are located approximately in a distance of the given radius r to the estimated sphere center are used for the sphere fitting. To get a first estimation for the sphere center, at first the point cloud of the tof coordinate system is transferred to a two dimensional plane, where the (x, y) remain and the distance information is represented by a gray value. This is just an ordinary 2D image, which can be used in a standard image processing algorithm.

3.2.1 Hough Circle Transform

Sonka [17] suggests the Hough transform to detect circular objects in an image. The line detecting Hough transformation (like in Gonzalez [18]) would suggest to use a three dimensional voting matrix to detect a circular object. Two dimensions for the x and y coordinate of the circle center and one for the circle radius r. An implementation of the circular Hough transformation could work like this. The original image is transformed to a binary edge image. Every edge pixels votes for the circle centers, for all possible x, y and r values, for which it could be a circle point (figure 3.4). The entry in the matrix with the most votes determines the variables x, y and r.

The implementation of OpenCV works a little different. To avoid the memory and time consuming three dimensional voting matrix, it votes only for the circle center coordinates (x, y) and determines the radius r for the most probable center (x, y) after the voting has been completed. To vote for a circle center the algorithm determines

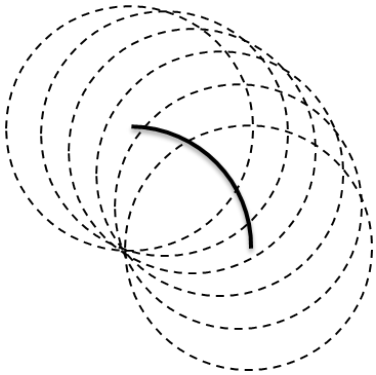


Figure 3.4 Hough voting (dashed) with a fixed radius r for a circular segment (solid line).

at first the angle of the potential circle and votes along a straight line orthogonal to the circular arc [19]. The OpenCV implementation of the circle detection does not yield a stable solution for the tested data, since the images of the table tennis balls have a too low resolution to contain enough border pixels to vote along their orthogonal line. A Hough transformation with an ordinary three dimensional voting matrix should work fine. But instead another simple algorithm, which is described in the next section 3.2.2 , has been implemented.

3.2.2 Local Minima Search

Since an algorithm to fit a sphere with known radius r is needed anyway for the sphere fitting (section 3.2.3), a simple algorithm has been designed to detect the center of the sphere, which will be used as a start value for the optimization algorithm. If you look at a sphere, then the center of the sphere is just r mm behind the closest spherical surface point (figure 3.5), when r is the radius. The point with the smallest distance z to the camera is the closest point. To allow several spheres in an image, a local neighborhood must be considered. To determine this point, it is just necessary to erode [17] the 2D distance image with a large kernel. Every z value which is equal to the eroded pixel with the same coordinate is a local minima for a neighborhood, with the same size as the kernel. In general the local minima is due to the noisy measurement not situated at the circular center (figure 3.6), which is treated by several iterations of the sphere fitting, which is described in the following sections.

3.2.3 Sphere Fitting

Not all pixels obtained by the camera can be used for the geometrical fitting, so only the pixels which are part of the spherical surface must be segmented. To do this, the sphere candidate from the 2D image is transformed back to its original 3D coordinate. All pixels which are closer than the radius plus a tolerance of 40 mm to the candidate

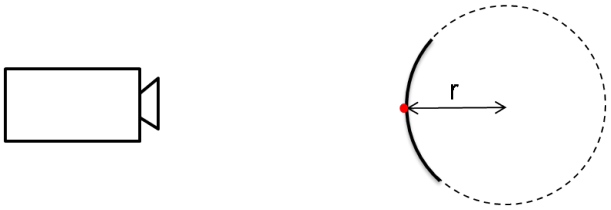


Figure 3.5 The closest point of the sphere to the camera is just r mm in front of the spherical center.

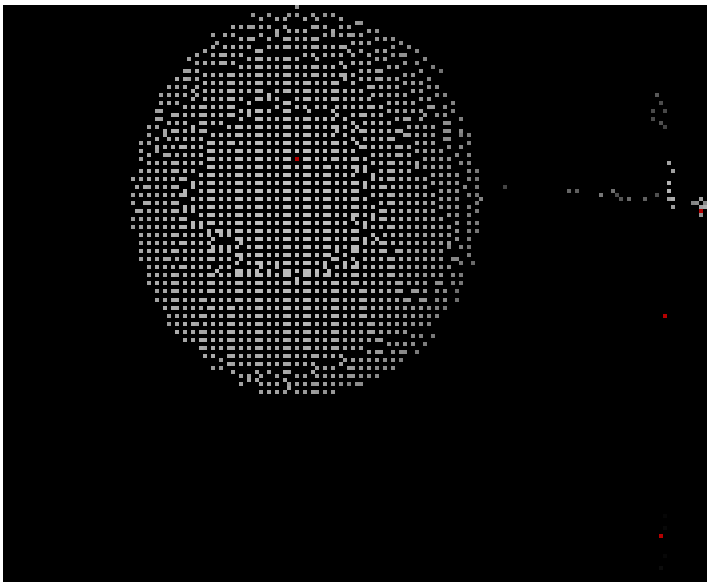


Figure 3.6 Gray pixel represent the distance, while red pixels are a local minima.

centers (a, b, c) are used for the fitting algorithm. If the candidate is the center of a sphere, then the surrounding pixels is the observed spherical surface.

The segmented surface pixels (x_i, y_i, z_i) are now used to determine a new estimation of the sphere center (a, b, c). The distance $d_i(x, y, z)$ is the distance of each pixel (x_i, y_i, z_i) to the spherical center (a, b, c). The objective function $f(a, b, c)$ is defined by the squared sum of the differences of the radius and the pixel distance to the center $d_i(x, y, z)$. To minimize the objective function $f(a, b, c)$ the Levenberg-Marquardt algorithm [20] is used. The stop criterion is a small error with at most 300 iterations. For chosen examples there was a clear global minimum for $f(a, b, c)$ without a risk for ending at a local minimum. More about optimization and the corresponding objective is described in the more general case of arbitrary surface matching in section

5.2.3.

$$\begin{aligned} & d_i(x,y,z) && = \sqrt{(x_i-a)^2+(y_i-b)^2+(z_i-c)^2} \\ \min \quad & f(a,b,c) && = \frac{1}{n}\cdot\sum_{i=1}^{n}\left(\sqrt{(x_i-a)^2+(y_i-b)^2+(z_i-c)^2}-r\right)^2 \end{aligned}$$

3.2.4 Candidate Filtering

If some local minima have the same z coordinate they might lie close together. So, if a candidate has a distance of less than r to already found candidates, it is skipped. So if we are looking for several spheres in one frame than there is at most one candidate pixel for each sphere. For all remaining candidates, a set of surface pixels is created and fitted with a sphere (see section 3.2.3). The new coordinates (a, b, c) of the candidates are used for a second iteration. The surrounding pixels are extracted again and the objective function minimized. This is necessary since the first candidate coordinates are usually not in the spherical center due to the noise in the acquired data.

The candidates with the lowest value for the objective function are selected to be the observed spheres. Nevertheless, objects with only a few pixels tend to have a lower error sum than the sphere. Therefore, all candidates which have less than a thresholded amount of surface pixels are removed from the candidate set before the selection.

3.2.5 Iterations to the Final Coordinates

The selected surface pixels, which are used for the spherical fitting, have a strong influence on the calculated sphere position. On one hand, as many pixels as possible should be used to minimize the error. On the other hand, pixels which are part of the objective function but do not belong to the sphere in reality, lead to a bias to their position. To filter out outliers, only 95 percent of the pixels which are closest to the estimated spherical surface are used (compare figure 3.8). Several iterations of the Levenberg-Marquardt algorithm are used to determine the sphere position and every time the surrounding pixels are newly determinated since they might change for a new estimated spherical center. This continues until the position of the sphere moves less than 0.1 mm.

Finally the summary of the complete algorithm is displayed in figure 3.7. Furthermore the algorithm is presented in the following abstract code with a simplified program structure. Note that all distance and coordinate values are in mm.

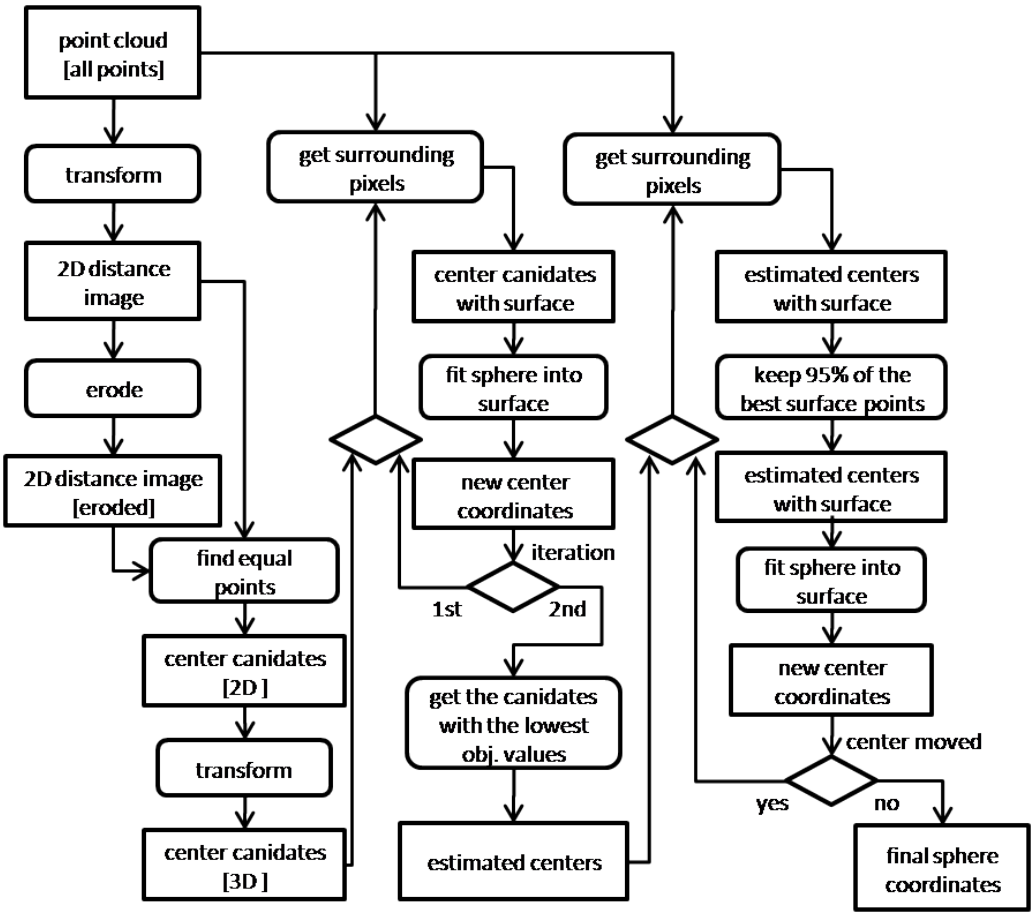


Figure 3.7 Diagram of the sphere fitting algorithm.

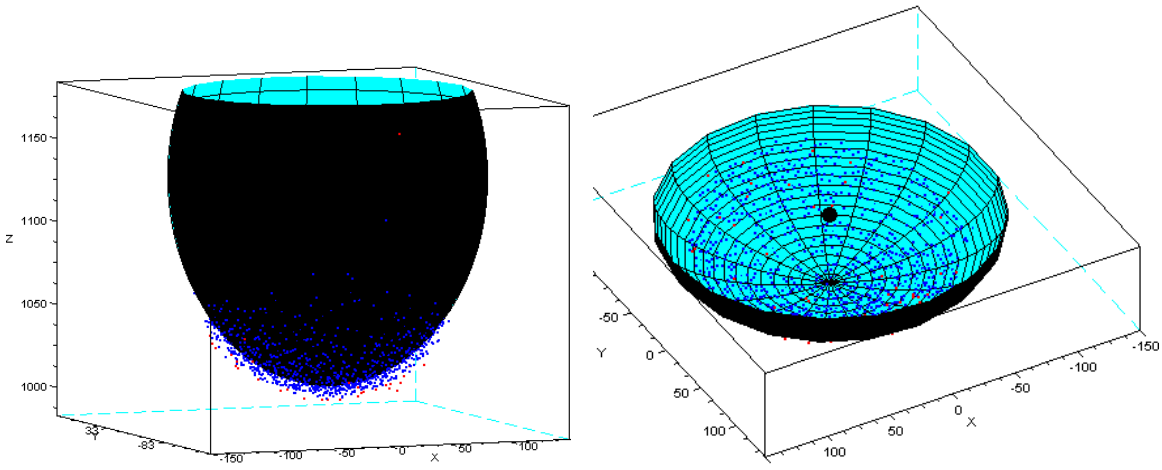


Figure 3.8 Sphere (black center point) is fitted into the selected pixels (green) with the removed percentile (red).

Algorithm 3.2.1: EXTRACTSPHERECENTER(p, r, n)

---

comment: Extracts n centers of spheres with radius r from the point cloud p

$(image2d, mapTo3D) \leftarrow$ convertTo2D(p)
$minImage \leftarrow$ erodeWithLargeKernel$(image2D)$
$possibleCenter \leftarrow$ Emtpy
for each $(x, y) \in image2d$
do { **if** $minImage[x, y] = image2d[x, y]$
  if distance3D(mapTo3d$(x, y), possibleCenter) < r$
    $possibleCenter$.add(mapTo3d(x, y))
for each $pC \in possibleCenter$
do { pC.z $\leftarrow pC$.z $- r$
$surfacePoints \leftarrow$ getPointsCloserThan$(pC, r + 50)$
$pC \leftarrow$ fitSphere$(SurfacePoints)$
for each $pC \in possibleCenter$
do { $surfacePoints \leftarrow$ getPointsCloserThan$(pC, r + 50)$
if $(surfacePoints$.size $< minPixel)$
  then remove(pC)
  else $pC \leftarrow$ fitSphere$(SurfacePoints)$
Keep only n possibleCenter points, with the lowest error in the fitSphere optimization
for each $pC \in possibleCenter$
do { **while** Sphere Center is Moving
do { $surfacePoints \leftarrow$ getPointsCloserThan$(pC, r + 50)$
$SurfacePoints \leftarrow$ KeepBestQuartile$(pC, SurfacePoints, 0.95)$
$pC \leftarrow$ fitSphere$(SurfacePoints)$
return (pC)

For the phantom with four visible spheres (figure 3.1) it is easy to identify which sphere is A, B, C and D. The longest distance between the spheres is $\overline{AD}$ and the shortest is $\overline{CD}$. So it is just necessary to extract the sphere which is closest and farthest away from the neighboring spheres at the same time to get D. A is the remaining with the farthest distance, C with the shortest distance and B must be the sphere is left over.

3.3 Result

All tests, which were done with spherical objects and a single camera, are published in this section. Two phantoms were used, the four table tennis balls (figure 3.1) and the styrofoam sphere (figure 3.2). All measurements were done with a camera with at least one hour time to heat before the measurement to run the camera with a constant temperature.

3.3.1 Static Observation

Tests, when neither the camera nor the phantom has moved are presented in this section.

A sequence of 100 frames was acquired with the table tennis phantom at a distance of 1.1 meter. The standard deviation of a single sphere was around 0.7mm in the (x, y) coordinate and 0.8mm in the z distance. The distances between the spheres had an absolute error of circa 2mm and a relative error of circa 0.008mm. All with the premise, that the reference measurement underlies an absolute error of around 0.5mm.

Furthermore the standard deviation of sphere D was observed for a orthonormal coordinate system defined by the three other spheres with B as the origin and the axes $e_1 := \frac{\overline{AB}}{\|AB\|}$, $e_2 := \frac{w_2}{\|w_2\|}$ with $w_2 := \overline{CB} - \langle \overline{CB}, e_1 \rangle$, and $e_3 := e_1 \times e_2$. The standard deviation for the coordinate of D is 2.05mm in e_1, 2.22mm in e_2, and 1.72mm in e_3 direction.

All further measurements were done with the styrofoam Phantom, whose size is closer to the size of a human head, which would be one possible object for the final application. The plotted positions of the calculated center for a static observation are displayed in figure 3.9. The distribution seems reasonable, without clusters. The (x, y, z) coordinate over several frames had a standard deviation of 0.3mm in (x, y) and of 0.8mm in z direction.

The histogram in Figure 3.10 shows the distance of the surface pixels of one frame to the fitted sphere surface. Pixels, which had a distance of less than sphere radius plus 20mm to the estimated center of the sphere were used for the fitting and plotted in the histogram. The pixels are symmetrical distributed, which is no surprise since the sphere was fitted into this point cloud. Nevertheless the histogram looks convenient

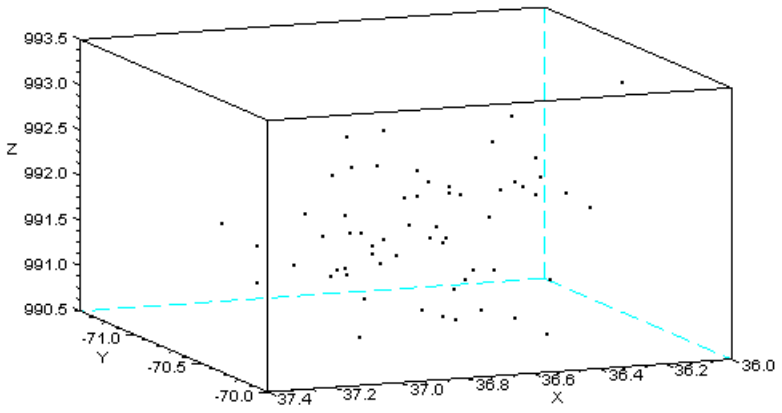


Figure 3.9 Position of the fitted centers of the spheres.

and shows the noise for a single pixels. The standard deviation of distance to the spherical surface is 4.1mm.

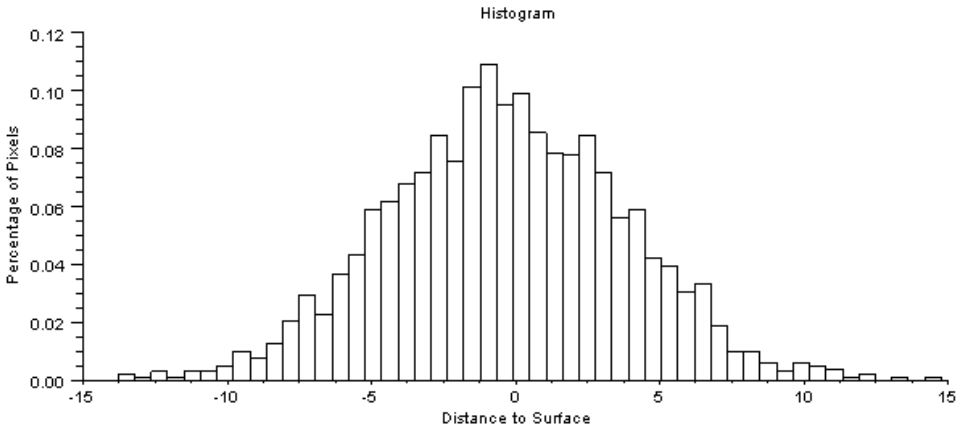


Figure 3.10 Distance of the measured surface points to the fitted sphere surface.

3.3.2 Shifted Sphere

After the studies of the random error for the measurement, the focus is now how exact a change in position can be determined. The styrofoam phantom was positioned 1m away from the camera and 80 frames were acquired. The phantom was moved 100mm closer to the camera and another 80 of frames were acquired. After a median filtering both positions were compared. For a median filter of size 80 the absolute error for

the 100mm shift was 0.5mm. For a median filter with a size of 4 the error is up to 1mm (figure 3.11) and for a smaller median filter even larger.

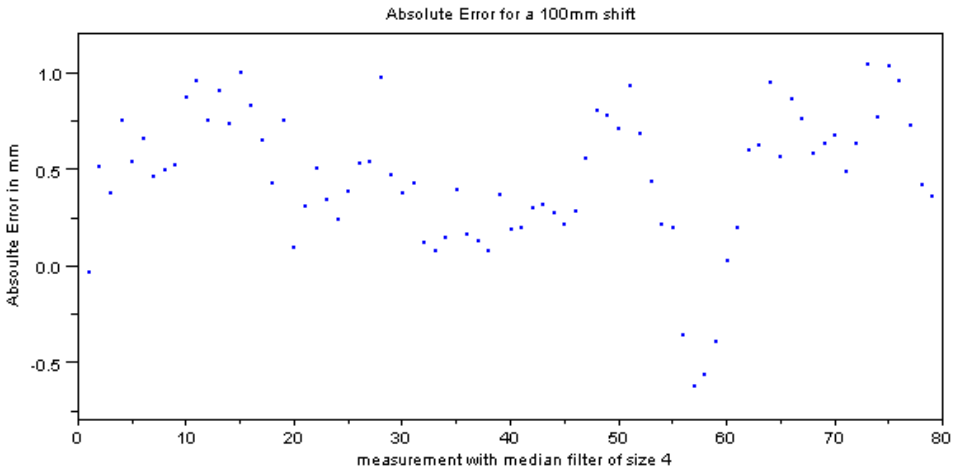


Figure 3.11 Measurement Error for a 100mm shifted sphere.

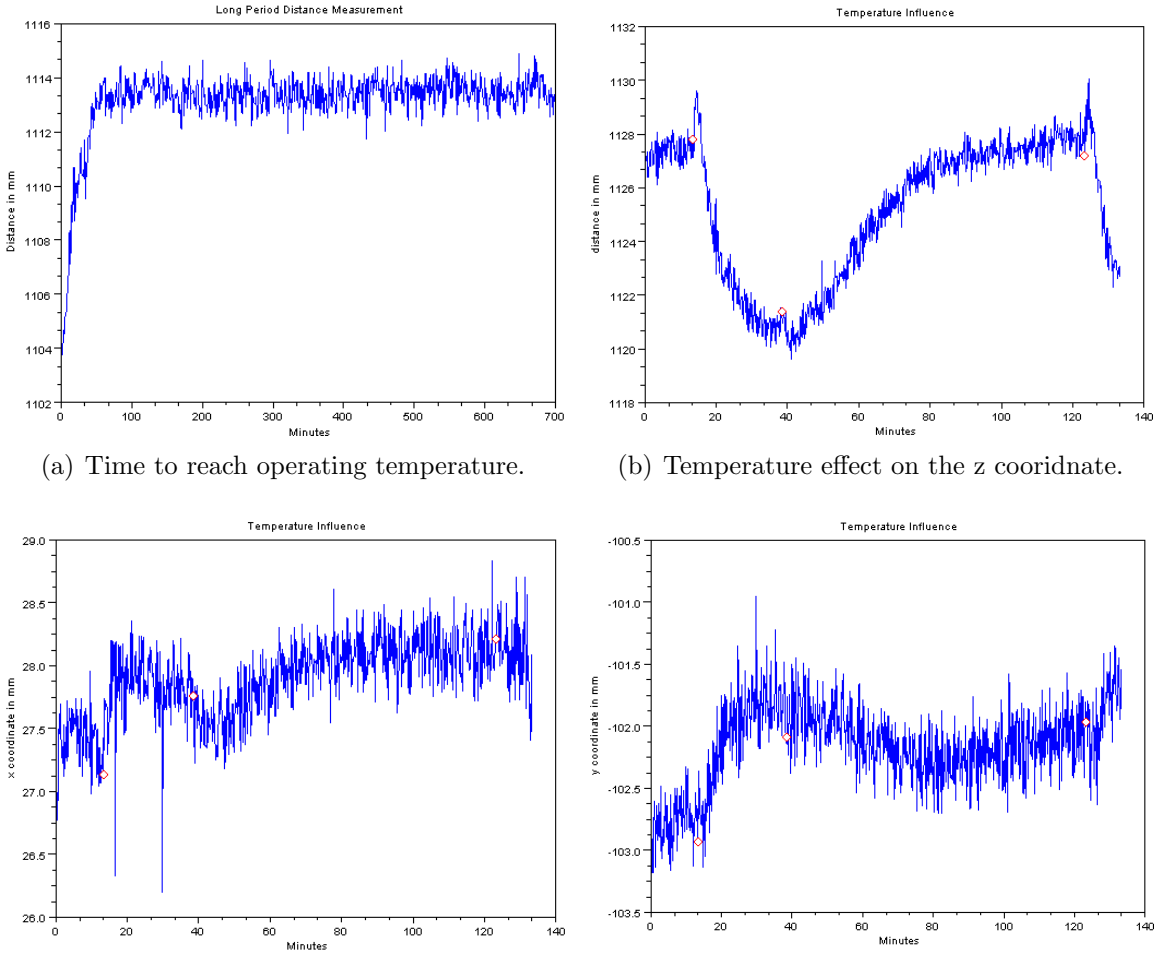
3.3.3 Temperature Influence

The temperature of the camera also has an influence on the measured values. The camera and the styrofoam phantom were set in a fixed position while the position were observed by the camera. For the first experiment (figure 3.12a) the camera was started with room temperature and heated itself up to circa 40°C. It took about one hour until equilibrium was reached. During this time, the calculated z coordinate changed 10mm before it stabilized. The coordinates x and y were only affected by a minor change.

In the second experiment about the influence of heat, the camera started with operating temperature (40°C) and got cooled by a small ventilator for certain time periods. The start and the end of the cooling phase are plotted with a red circle. With start of the cooling phase the measured distance increases at first by two millimeter and drops afterwards significantly by 7mm less than at the start (figure 3.12b). The jump of two millimeters might be explainable by the temperature sensor in the camera, which is used by the camera to correct the measured values. After the cooling stops, the distance value increases slowly to the original value. The process until the original distance value is achieved takes about 50 minutes. The influence to the x and y coordinates is with less than 2mm rather small compared to the z distance (figure 3.12c and 3.12d).

3.4 Discussion

The absolute accuracy does not matter so much. For the application is the relative vector of the object to the LINAC focus relevant. The accuracy of a static observation with a standard deviation of 0.3-0.8mm and an error of 0.5mm for a 100mm shift are quite promising and justify further studies. The error through multipath should not be a big problem in the therapy room, since the camera will be fixed at one position and the room inventory will change only slightly. Multipath on the patient skin might be a problem, which could be solved by using two cameras, such that at every relevant body part, one camera always has a clear view. Contrary, the temperature has a too strong influence on the camera to be able to ignore it. At first the camera can be supplied with power one hour before the therapy starts, to heat up to the working temperature. But the camera should also return feasible positions when the temperature in the room is changing. One solution would be with control engineering to keep the camera body temperature constant. Another one would be the use of two cameras, which is discussed in the next chapter, particularly in section 4.3.2.



(a) Time to reach operating temperature.

(b) Temperature effect on the z cooridnate.

(c) Temperature effect on the x cooridnate.

(d) Temperature effect on the y cooridnate.

Figure 3.12 Temperature influence for static observations.

Chapter 4

Stereo Camera System

What is better than a ToF camera? Two cameras? This chapter is about how two cameras can increase the accuracy for the patient positioning.

4.1 Experiment

The same Styrofoam sphere was used as in chapter 3 to measure the accuracy of the camera system as well as for the calibration. The cameras were fixed on an aluminum frame at top of a table. The angle between the cameras was about 50° degrees. Both cameras acquired independently positions of the sphere center with the algorithm of chapter 3. The next sections are about how those positions can be combined. Furthermore at the result section 4.4 are interferences between the cameras described.

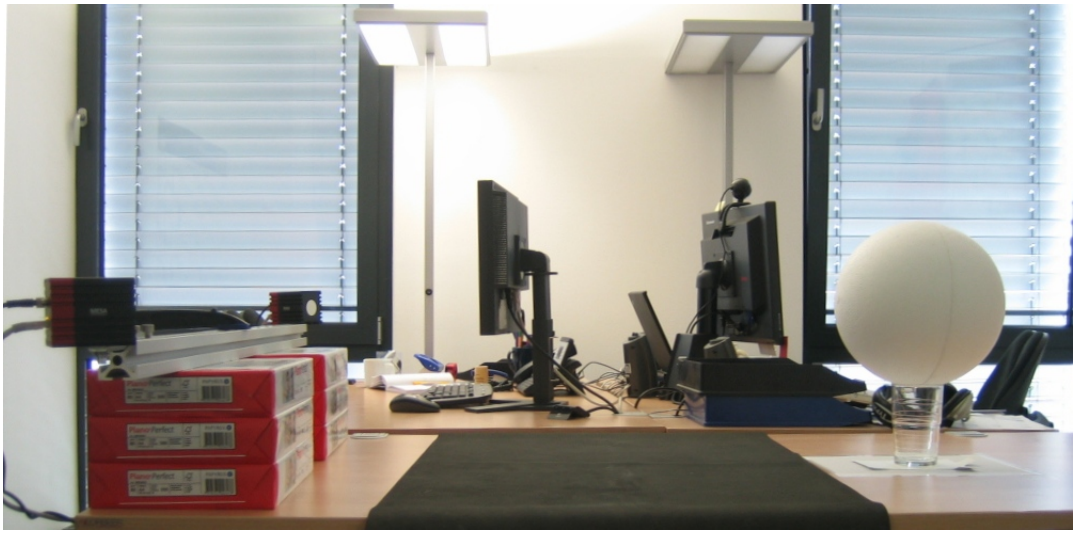


Figure 4.1 Stereo camera system on top of a paper stack recording the styrofoam sphere.

4.2 Transformation Matrix

To transform the a point $\overline{x}_1$ in the coordinate system of one camera to a point $\overline{x}_2$ in the coordinate system of another camera a transformation matrix T can be used $\overline{x}_2 = T \cdot \overline{x}_1$ (Bronstein [21]). For the three dimensional space there are six degrees of freedom for the transformation matrix $T(x, y, z, \alpha, \beta, \gamma)$, three translations x, y and z and the three rotations α, β and γ. While x, y and z are the translations along the axes, α, β and γ are the rotations around the axes. A translation cannot be expressed by a 3 by 3 transformation matrix. With extended coordinates $\overline{x} = (x, y, z, 1)$ the translation can be expressed by a linear transformation $\widetilde{T}$. The transformation matrix T can be split in several parts, since the matrix multiplication is associative. So we have the following matrix

$$T(x, y, z, \alpha, \beta, \gamma) = \widetilde{T}(x, y, z) \cdot R_x(\alpha) \cdot R_y(\beta) \cdot R_z(\gamma)$$

with

$$\widetilde{T}(x, y, z) = \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_y(\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z(\gamma) = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4.3 Algorithms

With help of the transformation matrix it is possible to transfer the coordinate system of one camera to the other camera system. In the next section 4.3.1 is described how to obtain such a transformation matrix. In the following section 4.3.2 is described how to combine the two coordinates of the sphere in the best way.

4.3.1 Calibration

For the calibration of the cameras, we need a transformation matrix T which allows us to transform the position of an object in the right camera system p_B to the position the same object hast in the left camera system p_A. It follows, that $p_A = T \cdot p_B$ plus a measurement error. With three known points (p_1, p_2, p_3) it is possible to calculate a unique transformation matrix T. The error g of the transformation should be as small as possible.

$$g = \sum_{i=\{1,2,3\}} (T \cdot p_{iB} - p_{iA})$$

To get the transformation matrix are three positions of the sphere acquired and the error of the calibration g is minimized to get T. To minimize the error of the calibration, the median position of 100 frames was used and the spheres were not placed inside a plane.

4.3.2 Combining the Positions

With help of T it is now possible to transform all coordinates from the left camera to the right camera. This should be used to increase the accuracy of the determination of the position of an object. Camera A has acquired the position p_A. Camera B has acquired the position $\tilde{p}_B = T \cdot p_B$ for the same object. Since each measurement contains errors, both positions are not equal. One possibility to half the error, would be to average over both positions $p = \frac{1}{2}p_A + \tilde{p}_B$. We know that the (x, y) coordinate is more accurate than the z coordinate of p_A and $\tilde{p}_B$ from section 3.3. With this premise p_S can be calculated as the position where the lines (A, p_A) and $(T \cdot B, T \cdot p_B)$ intersect (figure 4.2) with A the position of the left camera and B the position of the right camera. Since two lines are not intersecting in general in the three dimensional space we take p_S in the middle of the line which is the shortest distance between both lines.

4.4 Result

A ToF camera sends a pulsed light impulse out and when it comes back, it measures the phase of the light. Therefore other light sources might disturb the signal. Other light sources like the second camera. A minimal disturbance is achieved if both cameras are set to a different modulation frequency and an equal integration time. Nevertheless, there is a small disturbance measureable if both cameras are operating at the same time. For the experiment setup in figure 4.1 the standard deviation of the x coordinate doubled, while the y and z coordinate remained unchanged (compare section 3.3.1).

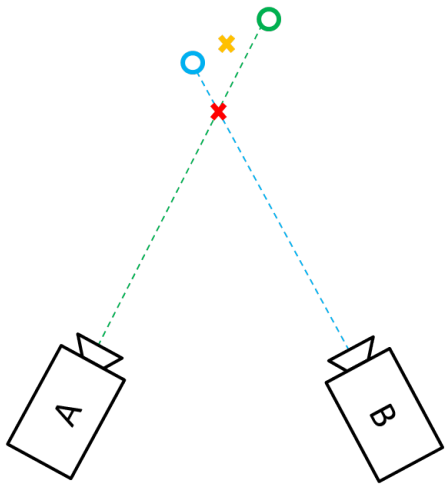


Figure 4.2 Possible combination of a stereo camera system.

Standard deviation [mm]		
x	y	z
1.3	0.7	1.0

The transformed position of an object in camera B should be equal to the position of the object in camera A. There are always errors present in measurements, such that the positions are not exactly the same. 100 frames of the Styrofoam phantom were taken and the positions p_A and $(\tilde{p}_B = T \cdot p_B)$ compared. The medium distance was 1.5mm and the largest distance 3mm.

The two points p_A and $\tilde{p}_B$ combined yield the position p_S. Figure 4.3 shows a test about the influence of temperature on p_S. Camera A and B were observing the Styrofoam sphere. After 5 minutes camera A got cooled for 10 minutes. The x coordinate of p_S varies in a range of 2mm and is only dependend on p_A and not on p_B. The y coordinate varies in range of 0.6mm and is an average out of p_A and p_B. The z coordinate varies in range of 3mm and depends only on p_B.

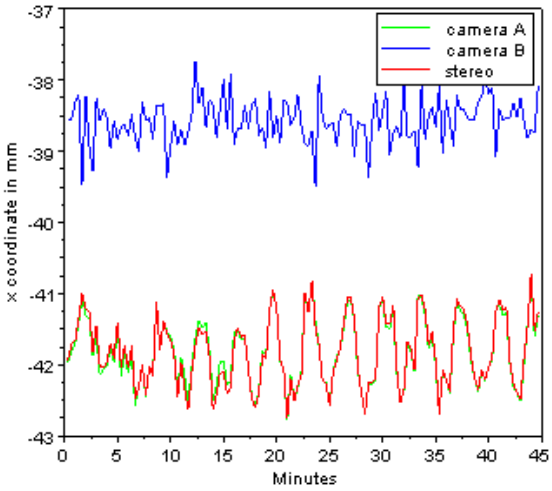
4.5 Discussion

The range of the z coordinate for the stereo camera system decreased to 3mm compared to 7mm with a single camera through cooling with the stereo camera. Contrary the x coordinate is not as accurate as with a single camera. This is explainable with the higher standard deviation for the cameras in the x coordinate due to interference between the cameras. The interference only in the x coordinate, is explainable since the cameras are positioned along their x-axis. To ensure this, tests with other camera positions could be done.

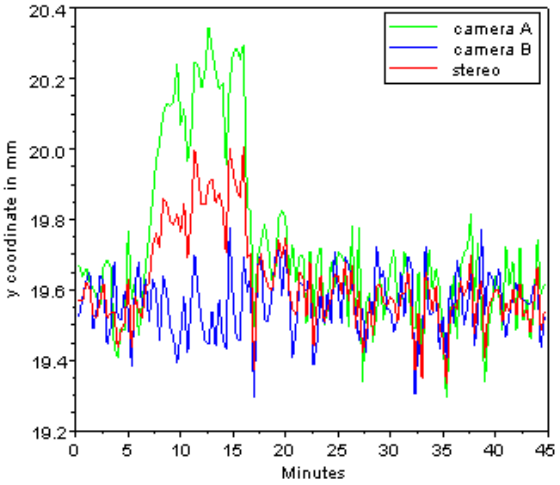
The strong dependency of the coordinates of p_S in figure 4.3 might be surprising at first glance. Actually it is exactly what was intended. At first, the z coordinate

of p_A is not used, instead the information of p_B is used. Secondly, the z coordinate of p_B is not used. The plot displays the coordinates in the camera system of camera A, or in other words p_B has been transformed with T. $\tilde{x}_B$ depends on z_B due to the camera positions. That means, that the distance information z is not used in this system, which is actually the big benefit compared to a regular RGB camera. Instead of a noisy ToF stereo system a stereo system with regular RGB cameras could be used. Only the detection of object without the distance information would be more difficult but the whole system could achieve a higher accuracy. One step back, the transformed position ($T \cdot p_B$) and p_A of the same object have a median distance of 1.5mm. This is quite high and might be due to the calibration. Several tests could not improve the results, neither considering more frames for determining the calibration positions (p_1, p_2, p_3) nor the use of more calibration points. If the calibration positions are all inside a plane, the results are a little worse.

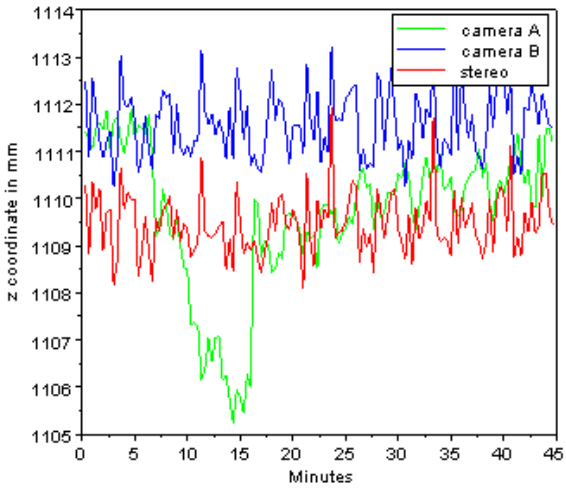
A smarter calibration might still yield in an increased accuracy to calculate the position of an object. Regardless, two cameras still increase the field of view, which is useful to recognize rotations and allows combining the information of two cameras for objects which have a shape which causes multipathing. How to calculate the rotation of a object between two images a registration of arbitrary point clouds is necessary, which is the topic of the next chapter.



(a) Temperature effect on the x cooridnate.



(b) Temperature effect on the y cooridnate.



(c) Temperature effect on the z cooridnate.

Figure 4.3 Temperature influence on the stereo ToF camera system in the coordinate system of camera A.

Chapter 5

Surface Matching

After the matching of spherical objects, it is time to handle the surface matching for arbitrary surfaces.

5.1 Experiment

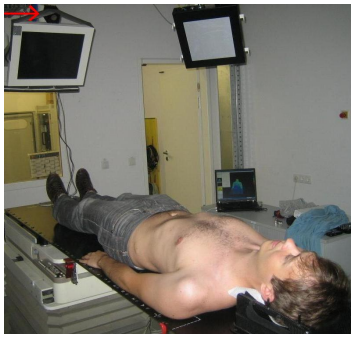
At first a reference image of a head dummy (figure 5.1a) was acquired at the distance of one meter. Afterwards, the dummy was moved a determined distance or rotated by 5°, 10° or 15° degrees.



(a) Head dummy on top of a rotation device.



(b) Torso dummy out of plastic.



(c) Registration setup for a human body. The ToF camera is marked with a red arrow.

Figure 5.1 Tests for the Surface Registration.

Another shift was simulated with a plastic phantom (figure 5.1b) at a camera distance of 1m and a angle of about 50 degrees from above. For the third setup a test person was lying on a couch with an easy breathing (figure 5.1c). The pixels which belong to the human surface could be easily extracted by using only pixels with high

amplitude. The black couch and objects in a large distance and bad reflectivity could be removed for the calculation. The electronically couch was moved a specific shift and compared with the calculated result to determine the accuracy of the registration. The calculation of the shift was done with the following algorithm.

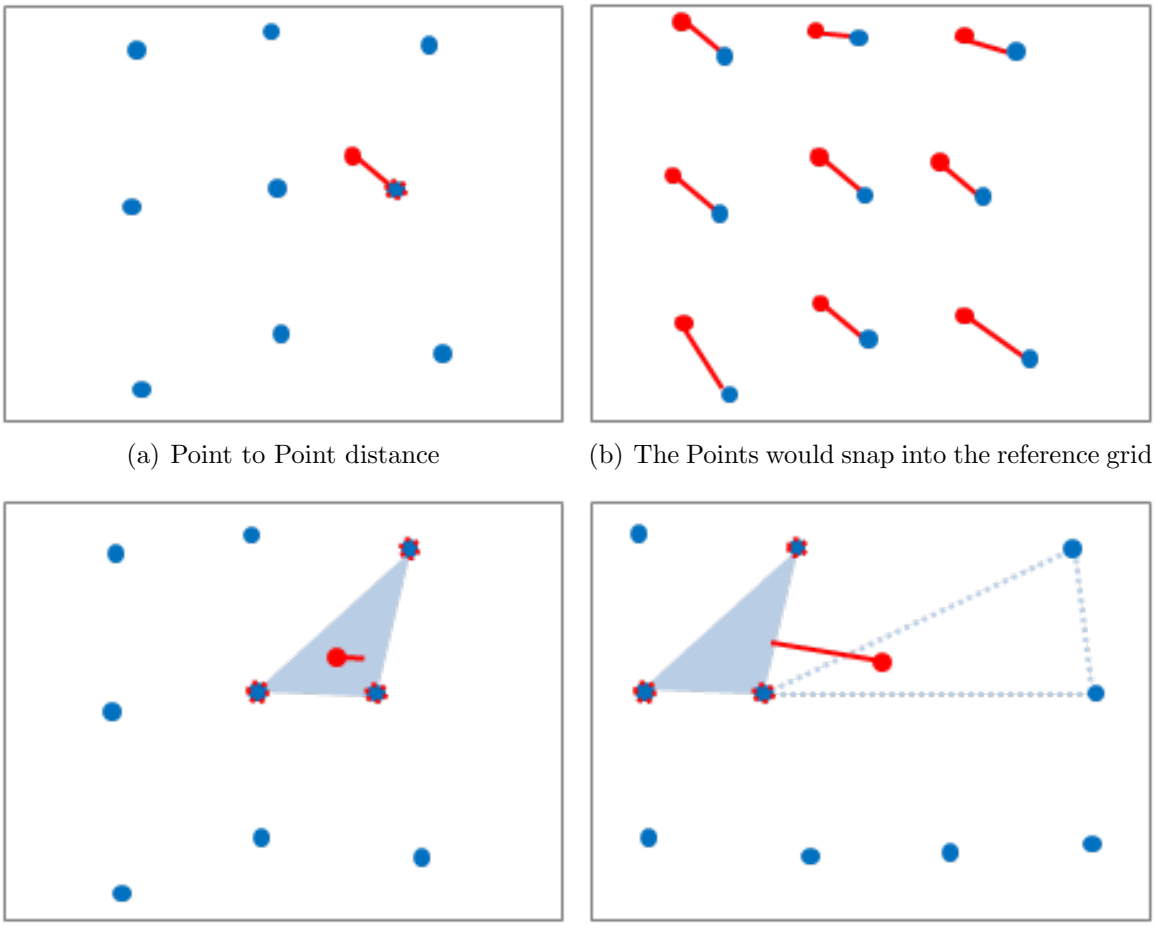
5.2 Optimization Algorithm

To determine the distance and rotation between two surface point clouds f_1 and f_2 a transformation matrix $T(\bar{x})$ must be calculated, such that the distance between surface f_1 and the transformed surface $(T(\bar{x}) \cdot f_2)$ is minimized. To do this, it is necessary to define a objective function $g(\bar{x})$, which describes the distance between the point clouds and to use an optimizer, which finds the best translation and rotation values $\bar{x}$ for the transformation matrix $T(\bar{x})$ to minimize $g(\bar{x})$. In section 5.2.1 the distance function $d(p, f)$ for a point p to a point cloud f is defined. In section 5.2.2 these distances are summarized to define the distance $d(f_1, f_2)$ between two point clouds f_1 and f_2 yield finally to an objective function $g(\bar{x})$. At last, in section 5.2.3 an algorithm is presented to calculate a transformation matrix $T(\bar{x})$ such that the objective function and therefore the distance between f_1 and $(T(\bar{x}) \cdot f_2)$ is minimized.

5.2.1 Point to Point Cloud Distance

There are several ways to define the distance $d(p, f)$ between a point p and point cloud f. The easiest distance function would be $d(p, f) = \|p - f(i)\|$, such that $f(i)$ is the closest point of the point cloud f to the point p (figure 5.2a). The only problem is, that the space between the pixels of the point cloud is empty, although the surface of the real object is there. Since the acquired pixels are approximately arranged on a grid, all the pixels p would snap to the reference grid f (figure 5.2b) and no subpixel accuracy could be achieved.

To fill the space between the pixels for the modeled distance a linear approximation can be used. For each pixel p, the three closest pixels $f(i)$, $f(j)$ and $f(k)$ are computed and the distance $d(p, f)$ is defined by the distance of p to the triangular subspace $t(i, j, k)$ determined by $f(i)$, $f(j)$ and $f(k)$ (figure 5.2c). In some cases it is possible that some grid pixels are missing. This might be the case, because of a surface with bad reflectivity or because the point is hidden by another object in front, for instance the nose can cover a part of the cheek. In this case the three closest neighbors would not interpolate the correct surface $t(i, j, k)$ and yield in an error (figure 5.2d). Nevertheless this simple distance function is used for the algorithm due to limited time and the possibility to manually avoid these bad cases in test situations. For the final application a triangulation should be computed and the distance between p and the closest triangular subspace should be used. Lee and Schachter [22] describe how to compute a feasible Delaunay triangulation.



(a) Point to Point distance

(b) The Points would snap into the reference grid

(c) Point to Triangle constructed through the 3 closest points

(d) A bad case for the Point to Triangle distance.

Figure 5.2 Distance of a point (red) to a point cloud (blue).

5.2.2 Objective Function

The distance $d(f_1, f_2)$ of the first point cloud f_1 to the second point cloud f_2 can be defined by the quadratic sum of the distances $d(p_i, f)$ of the points of one cloud $p_i \in f_1$ to the other cloud f_2.

$$d(f_1, f_2) = \frac{1}{|f_1|} \sum_{p_i \in f_1} d^2(p_i, f_2)$$

Since the objective is to calculate the translation and rotation between f_1 and f_2 we compute the transformation matrix $T(\bar{x})$ with $\bar{x} = (x, y, z, \alpha, \beta, \gamma)$ (more about transformation matrices in section 4.2) which minimizes the distance between $(T(\bar{x}) \cdot f_1)$ and f_2.

$$\min_{\bar{x} \in \mathbb{R}^6} \quad g(\bar{x}) = \frac{1}{|f_1|} \sum_{p_i \in f_1} d^2\big(T(\bar{x}) \cdot p_i, f_2\big)$$

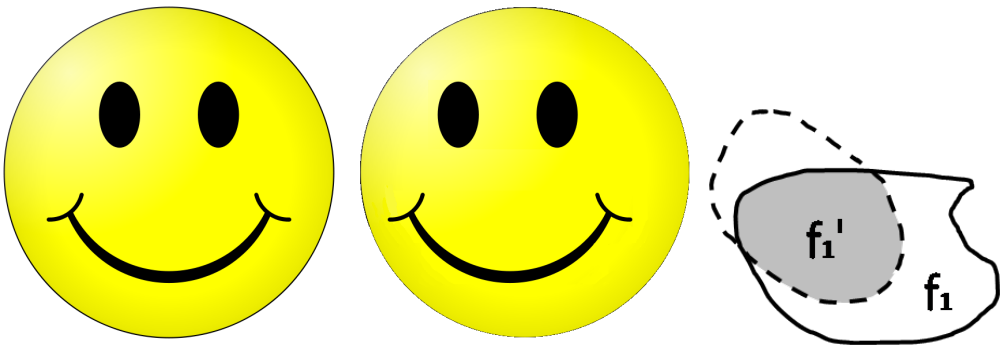
Only one side of the object is reflecting information to the camera, so there are no pixels on the backside of the object and a limited amount of pixels on the side. To allow an object rotation, the objective function needs to skip pixels which are only in the field of view of one frame f_1 or f_2. The smiley in figure 5.3(a-b) visualizes that it is not useful to match all pixels on the margin. For a correct matching it is more important to match eye to eye and mouth to mouth. To allow this constraint, all pixels of f_1, which are farther away than 7mm of f_2 are not cumulated to the distance sum, instead they increase the sum by a quadratic penalty factor $\left(\frac{|f_1|}{|f_1'|}\right)^2$ (compare figure 5.3c). The objective function becomes

$$\min_{\bar{x}\in\mathbb{R}^6} \quad g(\bar{x}) = \frac{1}{|f_1|}\left(\frac{|f_1|}{|f_1'|}\right)^2 \sum_{p_i\in f_1'} d^2\big(T(\bar{x})\cdot p_i, f_2\big)$$

with

$$p_i \in f_1' \quad \text{if} \quad d(p_i, f_2) < 7mm$$

and the distance function $d(p_i, f_2)$ from section 5.2.1, which is the distance of p_i to the triangular area constructed by the three closest points of f_2 to p_i.



(a) Smiley (b) Rotated smiley (c) f_1 to f_1' proportion

Figure 5.3

5.2.3 Approximation Algorithm

To keep the running time feasible, the optimizer should minimize $g(\bar{x})$ with as few calculations of $g(\bar{x})$ as possible. One good algorithm for this task is the UOBYQA (unconstrained optimization by quadratic approximation) algorithm by Powell [23]. It is also popular, since it is robust for noisy data. That is another benefit for the noisy data from a ToF camera.

UOBYQA optimizes the values $\bar{x} \in \mathbb{R}^n$ for a given objective function $g(\bar{x})$. The algorithm starts with a given starting value $\bar{x}_0$ and calculates the corresponding objective value $g(\bar{x}_0)$. For each dimension of $\bar{x}$ the algorithm changes one entry of $\bar{x}$ by

$+\rho_{\text{start}}$ and $-\rho_{\text{start}}$. For the changed values of $\bar{x}$, it calculates the new $g(\bar{x})$. That are $2n$ calculations of $g(\bar{x})$ for a dimension of $n = \text{Dim}(x)$. With these values, UOBYQA uses a quadratic approach to estimate a new vector $\bar{x}_i$ and continues to alter the entries by $\pm\rho$. This continues until $g(\bar{x})$ cannot be decreased any more by a step size of ρ and reduces therefore the step size ρ. The search is done until a before specified ρ_{end} is reached, where ρ_{end} must be less than ρ_{start}.

Back to the matching of the two surface point clouds f_1 and f_2. We look for the translation and rotation values $\bar{x}$ such that the transformation matrix $T(\bar{x})$ minimizes the distance of f_1 and $T(\bar{x}) \cdot f_2$ for the objective function $g(\bar{x})$. This calculation is implemented by the UOBYQA algorithm even though an approximation algorithm might result only in a local minimum. It is not possible to try every single possible value of $\bar{x}$ to minimize $g(\bar{x})$.

UOBYQA does not differ between the units of [mm] and [°] for the different variables, but it allows a scaling. The scaling has been chosen for rotation and translation: $s_\alpha = s_\beta = s_\gamma = \pi \cdot \frac{1}{180}$ and $s_x = s_y = s_z = 4$ [mm] such that 1° has the same weight as 4mm. The parameter ρ_{start} is according Powell supposed to be one tenth of the expected maximal change. A ρ_{start} of 1.0 is suitable for changes of the starting value x_0 for rotations up to 20° and movements up to 40mm.

5.2.4 Starting Value

A bad starting value $\bar{x}_0$ could lead to end up in a local minima at the end of the optimization. The difference $(\Delta x, \Delta y, \Delta z)$ between the centroids of the point clouds f_1 and f_2 is a good approximation for the translation $(x_0, y_0, z_0) \in \bar{x}_0$. The implementation of the starting value accelerated the calculation by 10% but changed the result for $T(\bar{x})$ only insignificantly.

$$(\Delta x, \Delta y, \Delta z) = \frac{1}{|f_1|} \sum_{(p_i) \in f_1} p_i - \frac{1}{|f_2|} \sum_{(p_i) \in f_2} p_i \qquad \text{with } p_i \in \mathbb{R}^3$$

The second moment can be used to get a second vector $(\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{z})$. Together with the centroid shift there are two vectors to estimate the rotations $(\alpha_0, \beta_0, \gamma_0)$.

$$(\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{z}) = \frac{1}{|f_1|} \sqrt{\sum_{(p_i) \in f_1} p_i^2} - \frac{1}{|f_2|} \sqrt{\sum_{(p_i) \in f_2} p_i^2}$$

5.3 Result

The results for rotations of the Styrofoam head are presented in table 5.1. The head was rotated around the y-axis. The nose moved from the middle to the right. With the help of a coordinate transformation, the origin was set to the center of the head to be able to compare the rotation values. The translation (x, y, z) is summarized with the Euclidian distance of the coordinates ($d = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$). Tests, acquired

case	translation [mm]	expected (β)	α	β	γ	difference (β)
1	1.2	5.0	-0.2	3.8	0.2	-1.2
2	1.3	-5.0	-0.4	-3.7	0.0	-1.3
3	2.5	10.0	0.7	7.5	0.0	-2.5

Table 5.1 Rotations for the Stryofoam phantom in degrees [°] with measured translation [mm]

case	expected	measured	difference
1	100.0	97.9	-2.1
2	50.0	37.2	-12.8
3	5.0	4.9	-0.1
4	5.0	4.7	-0.3
5	10.0	8.1	-1.9

Table 5.2 Translation for a plastic (case 1 & 2) and Stryofoam phantom (case 3-5) in [mm]

at 0.5m instead of 1.0m distance between camera and phantom improved the result only slightly.

Rotations were less than 0.5° for tests with pure translations. The difference for translation expermients with and without enabled rotations were less than 2mm. Since the rotation did not work that well, it was disabled. That way, rotation could not be the reason to be the error source in the later measurement and the degrees of freedom were reduced to three. The results for translation tests of the Styrofoam head and the plastic body are presented in table 5.2. The phantom was moved a certain distance with help of a scaled paper. The difference between the real and the measured distance are given for the Euclidian distance.

For the next test, the test setup was like in figure 5.1 with a patient lying on the couch while the ToF camera is acquiring images. The couch was moved electronically a specific distance and compared with the measured result. There are some tests which worked fine, and two tests with an error of more than 10 mm. In the table below the results of seven tests with the Euclidian distance are displayed.

Figure 5.4 shows (x, y, z) of the translation vector for a breathing patient as in the test setup figure 5.1. The (x, y) translation for registration is mostly lower than 0.5mm. Due to breathing the z translation moves up and down in a range of 4mm, which might got boosted by the noise ratio, which is strongest in the z coordinate (section 3.3.1).

case	expected	measured	difference
1	11.5	14.2	2.7
2	27.6	16.8	-10.8
3	27.0	21.9	-5.1
4	4.0	2.4	-1.6
5	11.0	10.7	-0.3
6	12.0	10.7	-1.3
7	64.3	51.7	-12.6

Table 5.3 Translation for a human body in [mm]

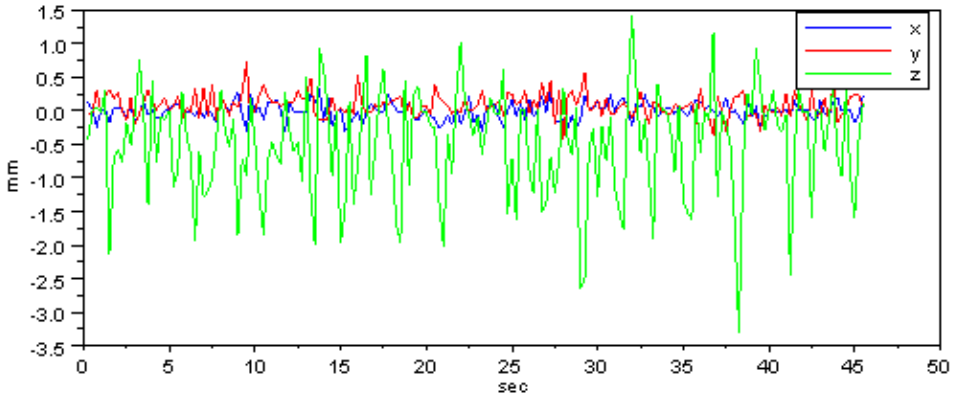
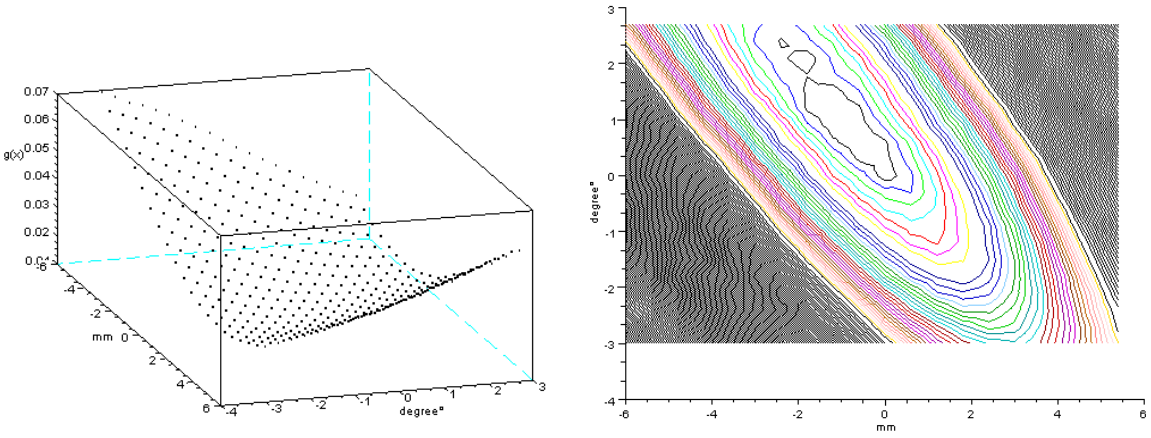


Figure 5.4 Registration of a silent lying and breathing human.

5.4 Discussion

The experiments for the registration of human body parts are partly very good and partly too bad for later applications. The calculated rotation angle with an error of 20% is not acceptable. The translation achieved some good results, especially for shifts in a short distance. But there are also some outliers, which must be investigated and excluded in later tests. Due to time limitation this could not be achieved in this thesis. Still, here are some ideas presented which can improve the result and some reasoning to justify parts of the algorithm.

The plots of $f(\bar{x})$ let assume, that UOBYQA finds the global minimum or at least a local minimum close to the global minimum. Since it is difficult to visualize a 6 dimensional space, in figure 5.5 are only the variables x and β, where x is the translation along the x-axis and β is the rotation around the y-axis displayed for a registration of an β=10° rotated head. The found optimum is at the coordinate $(x, \beta) = (0, 0)$. (x, β) are the most correlated variables and plots for other pairs of variables show a more clear valley for $g(\bar{x})$ than the (x, β) plot.



(a) Calculated points. (b) Contour plot

Figure 5.5 $g(\bar{x})$ with modified values for x and β.

The point clouds of a body registration are visible in figure 5.6. f_1 is cyan, $(T(\bar{x}) \cdot f_1)$ is blue and f_2 is red, such that $g(\bar{x})$ is minimized. The registration looks reasonable, since f_1 was shifted towards the right arm. Neverthenless, the calucated shift was only 21.9mm instead of 27mm (case 3 in table 5.3).

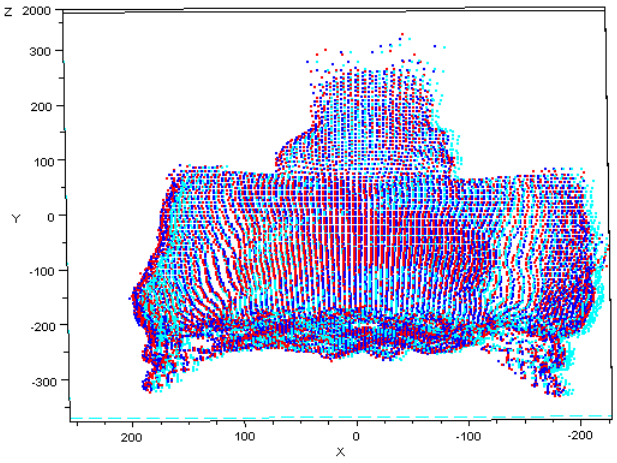


Figure 5.6 Point clouds for a shift towards the right arm.

There are several promising improvement undone which can significantly improve the final result. Like the start value for rotation $(\alpha_0, \beta_0, \gamma_0)$ (section 5.2.4), which is not implemented so far. Furthermore the penalty factor of the objective function, which leaves pixels unmatched if they have distance of more than 7mm (section 5.2.2). It filters successfully outliers out, but it did not succeed in leaving border pixels unmatched, as it would be necessary for a rotation (compare the smiley figure 5.3(a-b)). Also a better result could be obtained by removing some of the matched pixels before they are used for the matching. The pixels at the object border are noisy and maybe not visible at all at the other frame. So a better match is possible without them. In figure 5.7, only the pixels in the red frame could be used to match with another point cloud, instead of all pixels.

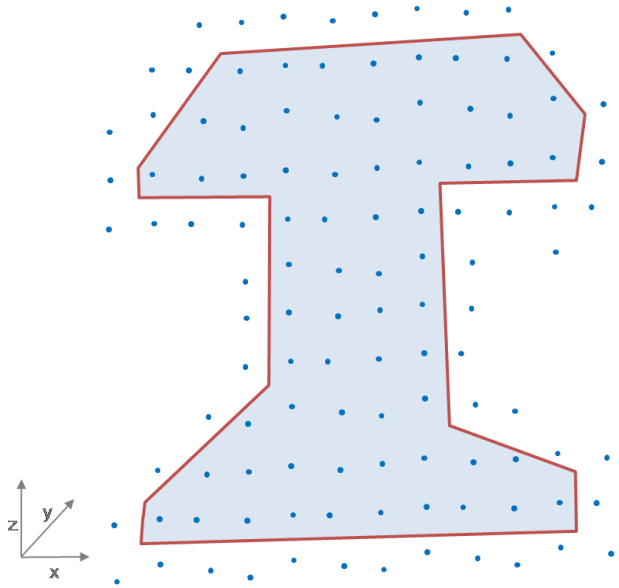


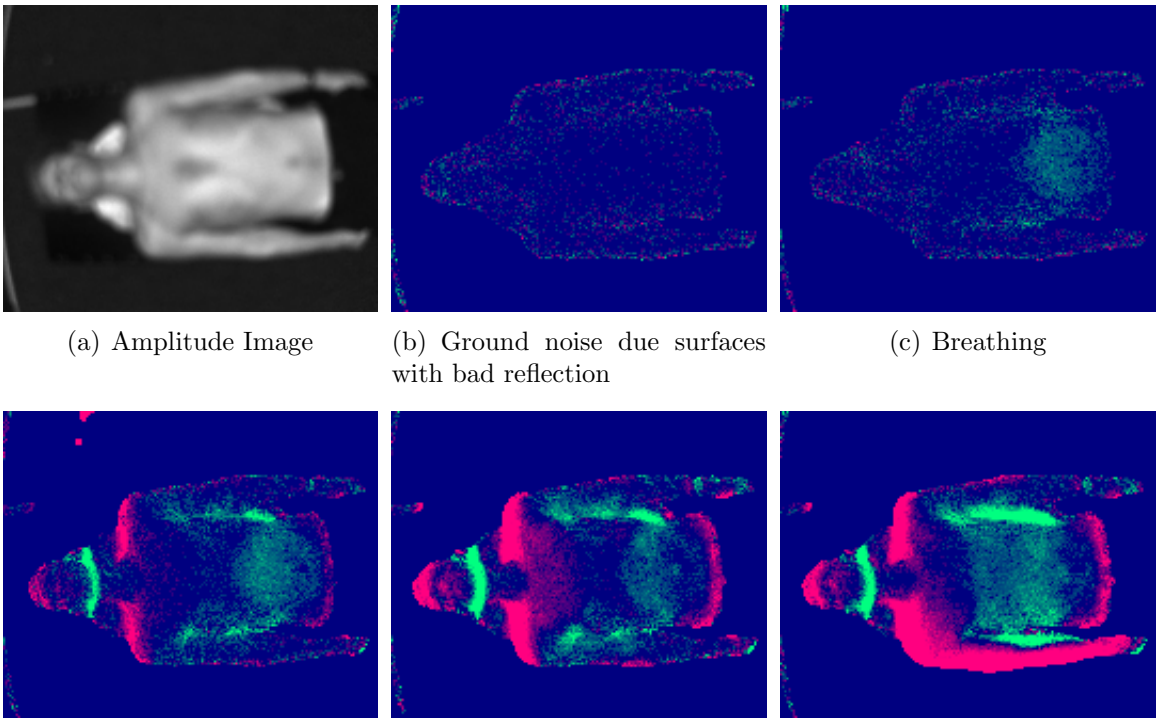
Figure 5.7 Removal of margin pixels for a better matching.

Chapter 6

Patient Surveillance

One application which is easy to realize with the help of a ToF camera is the observation of a patient. Just before the treatment starts, a reference image is taken. Afterwards, the live stream visualizes movements of the patient to the clinical personal. The clinical personal is now able to stop the treatment in case the patient moved out of the LINAC focus.

To do so, the camera is mounted above the patient. The distance values for each pixel are subtracted from the reference image and displayed in green and red (figure 6.1). Green in this case means, that the pixel is closer to the camera than in the reference image. Pixels, which are farther away, are red. To segment the patient, only pixels are taken, which have either in the reference or in the live image an amplitude value higher than a threshold of 3000. The background is black and has therefore a low amplitude. Furthermore this filters out a big part of the noisy pixels. Additionally a improved version of the surface registration (chapter 5) can display the movements in digits.



(a) Amplitude Image

(b) Ground noise due surfaces with bad reflection

(c) Breathing

(d) 10mm body shift towards the head

(e) 20mm body shift towards the head

(f) Additionally lifted arm

Figure 6.1 Pictures from an observation video.

Chapter 7

Conclusion

The ToF camera is able to register translations of large spherical objects with a medium error less than 1mm. Together with a frame rate which enable real time applications with a resolution of several thousand pixels the camera is very promising for new patient monitoring systems in the field of radio therapy. The visualization of the patient movements worked fine. Such an application, which makes an emergency stop as soon as a patient makes an unexpected movement, is realizable. A gating system is also possible, a system which pauses the treatment as soon as the breathing moves the tumor out of focus. Only the influence of temperature fluctuations must be controlled by a camera case with constant temperature. For certain positioning of the patient, the registration of the camera for some steady inner body parts is yet not precise enough. An X-ray scan matched with a CT-image is more precise. It has an error less than 1mm. On the contrary, a sub millimeter precision is useless for moving body parts. So a ToF camera with a high frame rate is more suitable for treatments with a tumor in a moving body part. It is not possible to state the accuracy of the registration of arbitrary surfaces in figures yet. There are some good results, but also too many outliers. At Brainlab, further research will be done in order to improve the registration algorithm. Furthermore, the developers of the cameras are creating new models, which are getting cheaper and more precise in order to improve the results. In general, the ToF camera is a very promising technology which could be used in a large variety of application and open new application fields for image analysis.

Bibliography

[1] W. Dörr, *Klinische Strahlenbiologie - kurz und bündig* (Urban & Fischer bei Elsev, 2006).

[2] Brainlab, "Radiosurgery," http://oncology.brainlab.com/ (Accessed Januar 25, 2010).

[3] VisionRT, http://www.visionrt.com/ (Accessed Januar 25, 2010).

[4] M. Krengli, S. Gaiano, E. Mones, A. Ballarè, D. Beldì, C. Bolchini, and G. Loi, "Reproducibility of patient setup by surface image registration system in conformal radiotherapy of prostate cancer," Radiation Oncology 4:9 (2009).

[5] Mesa, "Knowledgebase," http://www.mesa-imaging.ch/ (Accessed Januar 25, 2010).

[6] C. Schaller, A. Adelt, J. Penne, and J. Hornegger, "Time-of-flight sensor for patient positioning," In *Proceedings of SPIE*, 7258 (2009).

[7] M. Frank, M. Plaue, H. Rapp, U. Köthe, B. Jähnea, and F. A. Hamprecht, "Theoretical and Experimental Error Analysis of Continuous-Wave Time-Of-Flight Range Cameras," Optical Engineering 48 (2009).

[8] H. I. T. Kahlmann F. Remondino, "Calibration for increased accuracy of the range imaging camera swissranger," In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS Commission V Symposium, volume XXXVI*, pp. 136–141 (2006).

[9] S. Schuon, C. Theobalt, J. Davis, and S. Thrun, "LidarBoost: Depth superresolution for ToF 3D shape scanning," In , pp. 343–350 (IEEE, 2009).

[10] C. Beder, B. Bartczak, and R. Koch, "A Combined Approach for Estimating Patchlets from PMD Depth Images and Stereo Intensity Images," In *DAGM-Symposium*, pp. 11–20 (2007).

[11] P. Khongsab, "Signal Processing and Performance Evaluation of a PMD Camera for Space Docking," Master thesis (Luleå tekniska universitet, 2009).

[12] F. Tombari, L. Stefano, S. Mattoccia, and A. Zanetti, "Graffiti Detection Using a Time-Of-Flight Camera," In *ACIVS '08: Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 645–654 (Springer-Verlag, Berlin, Heidelberg, 2008).

[13] S. May, B. Werner, H. Surmann, and K. Pervölz, "3D time-of-flight cameras for mobile robotics," In *IROS*, pp. 790–795 (IEEE, 2006).

[14] S. Hsu, S. Acharya, A. Rafii, and R. New, "Performance of a Time-of-Flight Range Camera for Intelligent Vehicle Safety Applications,", 2006.

[15] PrimeSense, "PrimeSense Supplies 3-D-Sensing Technology to Project Natal for Xbox 360," http://www.primesense.com/ (March 31, 2010; Accessed April 27, 2010).

[16] H. Rapp, "Experimental and Theoretical Investigation of Correlating TOF-Camera Systems," Diploma Thesis (University of Heidelberg, 2007).

[17] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 2 ed. (Chapman & Hall, 1998).

[18] R. C. Gonzalez and R. E. Woods, *Digital image processing* (Prentice Hall, Upper Saddle River, N.J., 2008).

[19] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library* (O'Reilly, Cambridge, MA, 2008).

[20] K. Levenberg, "A method for the solution of certain nonlinear problems in least squares," In *The Quarterly of Applied Mathematics 2*, pp. 164–168 (1944).

[21] I. Bronstein, K. Semendjajew, G. Musiol, and H. Mühlig, *Taschenbuch der Mathematik*, 5. auflage ed. (Verlag Harry Deutsch, Thun und Frankfurt am Main, 2001).

[22] D. T. Lee and B. J. Schachter, "Two algorithms for constructing a Delaunay triangulation," International Journal of Parallel Programming **9,** 219–242 (1980).

[23] M. Powell, "UOBYQA: unconstrained optimization by quadratic approximation," Mathematical Programming **92,** 555–582 (2000).