Examensarbete

Kinect 3D Mapping

Examensarbete utfört i Datorseende vid Tekniska högskolan vid Linköpings universitet av

Anton Nordmark

LiTH-ISY-EX--12/4636--SE

Linköping 2012
Kinect 3D Mapping

Examensarbete utfört i Datorseende vid Tekniska högskolan vid Linköpings universitet av

Anton Nordmark

LiTH-ISY-EX--12/4636--SE

Handledare:  
Erik Ringaby  
ISY, Linköpings universitet

Johan Borg  
Saab Dynamics

Folke Isaksson  
Saab Dynamics

Examinator:  
Per-Erik Forssén  
ISY, Linköpings universitet

Linköping, 9 april 2012
3D kartering med Kinect

Anton Nordmark

This is a master thesis of the Master of Science degree program in Applied Physics and Electrical Engineering at Linköping University. The goal of this thesis is to find out how the Microsoft Kinect can be used as a part of a camera rig to create accurate 3D-models of an indoor environment. The Microsoft Kinect is marketed as a touch free game controller for the Microsoft Xbox 360 game console. The Kinect contains a color and a depth camera. The depth camera works by constantly projecting a near infrared dot pattern that is observed with a near infrared camera. In this thesis it is described how to model the near infrared projector pattern to enable external near infrared cameras to be used to improve the measurement precision. The depth data that the Kinect output have been studied to determine what types of errors it contains. The finding was that the Kinect uses an online calibration algorithm that changes the depth data.

Structured light sensor, Kinect
Abstract

This is a master thesis of the Master of Science degree program in Applied Physics and Electrical Engineering at Linköping University. The goal of this thesis is to find out how the Microsoft Kinect can be used as a part of a camera rig to create accurate 3D-models of an indoor environment. The Microsoft Kinect is marketed as a touch free game controller for the Microsoft Xbox 360 game console. The Kinect contains a color and a depth camera. The depth camera works by constantly projecting a near infrared dot pattern that is observed with a near infrared camera. In this thesis it is described how to model the near infrared projector pattern to enable external near infrared cameras to be used to improve the measurement precision. The depth data that the Kinect output have been studied to determine what types of errors it contains. The finding was that the Kinect uses an online calibration algorithm that changes the depth data.
# Contents

<table>
<thead>
<tr>
<th>Notation</th>
<th>vii</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Purpose and goal</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Limitations</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Related work</td>
<td>2</td>
</tr>
<tr>
<td>1.3.1 KinectFusion: Real-Time Dense Surface Mapping and Tracking</td>
<td>2</td>
</tr>
<tr>
<td>1.3.2 3D with Kinect</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>2 Kinect</strong></th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 RGB camera</td>
<td>3</td>
</tr>
<tr>
<td>2.2 NIR camera</td>
<td>3</td>
</tr>
<tr>
<td>2.3 NIR Projector</td>
<td>3</td>
</tr>
<tr>
<td>2.4 OpenKinect/Kinect technical specification</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>3 Calibration</strong></th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Camera model</td>
<td>7</td>
</tr>
<tr>
<td>3.1.1 Intrinsic parameters</td>
<td>7</td>
</tr>
<tr>
<td>3.1.2 Extrinsic parameters</td>
<td>8</td>
</tr>
<tr>
<td>3.1.3 Pinhole camera</td>
<td>8</td>
</tr>
<tr>
<td>3.1.4 Pinhole camera with lens distortion</td>
<td>9</td>
</tr>
<tr>
<td>3.1.5 Fisheye camera model</td>
<td>9</td>
</tr>
<tr>
<td>3.1.6 Fisheye with lens distortion</td>
<td>10</td>
</tr>
<tr>
<td>3.2 Projector model</td>
<td>10</td>
</tr>
<tr>
<td>3.3 Panorama calibration</td>
<td>11</td>
</tr>
<tr>
<td>3.4 Depth Calibration</td>
<td>11</td>
</tr>
<tr>
<td>3.5 Depth to world coordinates</td>
<td>13</td>
</tr>
<tr>
<td>3.6 Projector calibration</td>
<td>13</td>
</tr>
<tr>
<td>3.7 Projector calibration against external camera</td>
<td>19</td>
</tr>
<tr>
<td>3.7.1 Vertical line pattern</td>
<td>19</td>
</tr>
<tr>
<td>3.7.2 NIR camera distortion impact on depth</td>
<td>20</td>
</tr>
<tr>
<td>3.7.3 Projector pattern impact on depth mesurements</td>
<td>20</td>
</tr>
</tbody>
</table>
## CONTENTS

4 Physical setup .......................................................... 23

5 Validation
   5.1 Representation of a flat surface .................................. 25
   5.2 Algorithm to calculate the flat surface vector ................. 26
   5.3 RANSAC ................................................................. 26
   5.4 Angle between flat surfaces on one image ......................... 27
   5.5 Angle between flat surfaces from different images ............ 27
   5.6 Validation test .......................................................... 27

6 Results
   6.1 Auto calibration .......................................................... 29
      6.1.1 Auto calibration test ........................................... 29
   6.2 Kinect pattern model .................................................... 29
   6.3 3D model results .......................................................... 30

7 Conclusion
   7.1 Topics for future studies .............................................. 35
      7.1.1 Combined Stereo and structured light sensor .................. 35
      7.1.2 Combining multiple depth panoramas into one 3D model ... 36
      7.1.3 Vertical line detection and complex Residual error compen-
            sation ................................................................. 36

Bibliography ................................................................. 37
ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIR</td>
<td>near infrared</td>
</tr>
<tr>
<td>FOV</td>
<td>field of view</td>
</tr>
<tr>
<td>DSLR</td>
<td>digital single lens reflex (camera)</td>
</tr>
</tbody>
</table>
A Microsoft Kinect is a device that delivers both RGB images and depth images. It is marketed and sold as a game control to Microsoft Xbox 360. The basic idea behind this thesis is to better understand the performance and the limitations of the Kinect and how it can be used in an indoor mapping system.

1.1 Purpose and goal

This master thesis is a feasibility study for using Microsoft Kinect for an indoor 3D mapping system. There are three sub goals:

The first goal is to model the Microsoft Kinect projector pattern in order to evaluate the possibility to use an external NIR camera for higher performance. An external camera can improve performance in two ways; it allows a longer baseline between the camera and the projector and it can have a higher resolution.

The second goal is to use an external camera for the RGB image. The Kinect’s built-in RGB camera has low image quality compared to DSLR cameras. The sub goal is to calibrate the Kinect and an external DSLR camera to create an overlaid image.

The third goal is to evaluate if repetitive depth measurements give the same results. This is a prerequisite in order to make a calibration in order to optimize the Kinect’s performance.
1.2 Limitations

In this master thesis the Kinect is placed on a tripod during data collection, to get stationary position. This is done because the interface used to communicate with the Kinect cannot deliver RGB and NIR images at the same time. Artifacts from taking images on the move, such as motion blur and rolling shutter, which are outside the scope of this thesis, are avoided. The Kinect will be rotated on the tripod to simplify the motion estimation of the Kinect. No test have been done with hand-held cameras in this thesis.

1.3 Related work

The Microsoft Kinect has gained popularity in the academic community as a low cost depth camera that delivers reasonable performance. Several academic projects have been done with the Kinect, below there are two examples:

1.3.1 KinectFusion: Real-Time Dense Surface Mapping and Tracking

KinectFusion is able to perform real-time 3D reconstruction of a room sized scene. The data is collected using a hand held, relatively fast moving Kinect. To achieve real-time performance using 30 frames a second the calculations are performed on a Graphic Processing Unit. The resulting 3D model is much better than the individual depth frames making sub millimeter details visible [Kin].

1.3.2 3D with Kinect

The paper 3D with Kinect [Jan Smisek, 2011] focuses mainly on calibrating the Kinect. In this paper a procedure for calibrating the Kinect’s RGB and near-infrared (NIR) camera together are described. Methods for calibrating depth values are also described. Some of the calibration steps used in this master thesis comes from this paper.
The Kinect consists of a near-infrared (NIR) camera, an RGB camera and a NIR-projector. Additional features that are available in the Kinect are four microphones, tilt mechanism and accelerometers; these will not be used in this project. The Kinect works by emitting a fixed dot pattern of near infrared light which it observes with a NIR camera. The pattern offset is used to determine the distance to the observed environment.

### 2.1 RGB camera

The RGB camera has a resolution of 1280 x 1024 pixels for a 63 x 52 degrees field of view. The RGB camera delivers a medium quality image in indoor illumination, meaning that there is a high level of noise in the image. The noise level is reduced in better illumination, for example outdoors.

### 2.2 NIR camera

The NIR camera has a resolution of 1280 x 1024 pixels for a 57 x 47 degrees field of view. The NIR camera has less noise than the RGB camera [Widenhofer, 2010], this is due to a bigger CMOS sensor.

### 2.3 NIR Projector

The projector in Kinect consists of a NIR laser [Widenhofer, 2010] and two diffractive optical elements (DOE) [Alexander Shpunt, 2010]. The arrangement of the
Figure 2.1: Figure over how the Kinect projector works. A laser beam is projected at the first diffractive optical elements (DOE) that splits the beam into a pattern, the second DOE duplicates the pattern.

two DOEs and the laser is shown if figure 2.1. The patent application “Optical pattern projection” [Alexander Shpunt, 2010] explains how it works.

“The first DOE is configured to apply to the input beam a pattern with a specified divergence angle, while the second DOE is configured to split the input beam into a matrix of output beams with a specified fan-out angle.”

In the case of the Kinect, the pattern after the first DOE consist of 3681 dots arranged as shown in figure 2.2. The pattern is 180 degrees rotation invariant due to the greater ease of pattern manufacturing [Photonics, 2010]. In the case of the Kinect projector the matrix of output beams is a 3x3 matrix.

2.4 OpenKinect/Kinect technical specification

In this thesis the OpenKinect driver was used. Technical data from [wiki openKinect], the Kinect can deliver high resolution (1280 x 1024) RGB images at approximately 10 Hz frame rate or low resolution (640 x 480) RGB at frame rate of about 30 Hz. For high resolution (1280 x 1024) NIR images the Kinect has a frame rate of about 9 Hz and for low resolution (640 x 488) NIR Kinect has a frame rate of about 30 Hz. The depth data output has a resolution of 640 x 480 pixels, where the rightmost 8 pixel are always ”no data”, so the effective resolution is 632 x 480 pixels.
Low or high resolution RGB and depth can be collected at the same time. NIR images and RGB images cannot be collected at the same time. Low resolution NIR images and depth images can be collected at the same time.
In this chapter it is described how to calibrate the cameras and the Kinect projector. To do this the Kinect projector is modeled as a camera with some modification. First the pin hole camera model is described. Then four fish eye camera models are described. After the camera models have been described, a discussion about which camera model that is the most suitable to fit the Kinect projector follows.

3.1 Camera model

To perform any kind of measurements of the real world using images, the camera model has to be known. The camera model is a function that describes how the scene is mapped to the image. In Figure 3.1 the camera and world coordinates system are shown. The basic parameters can be divided into two groups, intrinsic and extrinsic parameters.

3.1.1 Intrinsic parameters

The intrinsic parameters are:

- Focal length.
- Camera sensor size.
- Lens distortion, measured as a radial distortion.
- Image plane displacement, measured as displacement from the optical axis.
3.1.2 Extrinsic parameters

The extrinsic parameters are:

- Camera location, described as location vector.
- Camera rotation, described as Euler angles.

3.1.3 Pinhole camera

The basic pinhole camera geometry gives the following relation between world coordinates $\hat{p} = (x_c, y_c, z_c)^T$ and camera coordinates $p' = (x_f, y_f)^T$. The camera is looking down the negative $z$ coordinates axis.

$$x_f = \frac{f \cdot x_c}{z_c} \quad y_f = \frac{f \cdot y_c}{z_c}$$  \hspace{1cm} (3.1)

To go from the $p' = (x_f, y_f)^T$ representation of a position on the image plane to a pixel representation $p = (U_i, V_i)$ the following formulas are used.

$$U_i = x_f \cdot \frac{w_1/2}{s_x/2} + w_1/2 \quad V_i = -y_f \cdot \frac{h_1/2}{s_y/2} + h_1/2$$  \hspace{1cm} (3.2)

$s_x$ and $s_y$ are the size of the of the sensor chip, measured in the same unit as $f$. $w_1$ and $h_1$ are the number of pixels in $x$ and $y$ direction respectively on the sensor chip. The camera sensor size and focal distance parameters are replaced with field-of-view side and field-of-view height parameters to achieve a more compre-
3.1 Camera model

3.1.3 Image of fisheye lens camera model. The world coordinate \( \hat{p} \) is projected at the image plane as a function of the angle \( \theta \).

\[
\begin{align*}
\frac{s_x}{2f} &= \tan(\text{fov}_s/2) \quad \frac{s_y}{2f} = \tan(\text{fov}_h/2) \\
U_i &= x_0 \times \frac{w_1/2}{\tan(\text{fov}_s/2)} + w_1/2 \quad V_i = -y_0 \times \frac{h_1/2}{\tan(\text{fov}_h/2)} + h_1/2
\end{align*}
\]

3.1.4 Pinhole camera with lens distortion

Real images usually suffer from lens distortion. Two forms of distortions are compensated for in the camera models that are used in this thesis, radial distortion and chip offset.

\[
r_0 = \sqrt{x_0^2 + y_0^2}
\]

\[
\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = (1 + k_2 \times r_0^2 + k_3 \times r_0^4 + k_4 \times r_0^6) \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}
\]

\[
U_i = x_1 \times \frac{w_1/2}{\tan(\text{fov}_s/2)} + w_1/2 \quad V_i = -y_1 \times \frac{w_1/2}{\tan(\text{fov}_h/2)} + h_1/2
\]

3.1.5 Fisheye camera model

The previous section described a pinhole camera with the addition of a distortion model. For wide angle lenses that approaches and sometimes exceeds 180 degree field-of-view different model is needed.

\[
r_c = \sqrt{x_c^2 + y_c^2} \quad \theta = \tan^{-1}(r_c/z_c)
\]
Fisheye cameras are angle-based cameras using a mapping function on the form \( r_i = R_f(\theta) \). Five different mapping function are described in [wikipedia, 2010]:

- **Equidistant** \( r_i = f \theta \)
- **Orthographic** \( r_i = f \sin(\theta) \)
- **Equisolid angle** \( r_i = 2f \sin(\theta/2) \)
- **Conform** \( r_i = 2f \tan(\theta/2) \)
- **Perspective** \( r_i = f \tan(\theta) \), same as pinhole camera.

This can be described as a camera model.

\[
U_i = \frac{x_c}{r_c} * R_f(\theta) * \frac{w_1}{s_x} + \frac{w_1}{2} \\
V_i = -\frac{y_c}{r_c} * R_f(\theta) * \frac{h_1}{s_y} + \frac{h_1}{2} \tag{3.10}
\]

The camera sensor size and focal distance parameters are replaced with field-of-view side and field-of-view height parameters to achieve a more comprehensible representation, in the same way as in the case of the pinhole camera. Rewriting Equation 3.9, and inserting Equation 3.3 into 3.10 gives:

\[
R(\theta) = \frac{R_f(\theta)}{f} \tag{3.11}
\]

and

\[
U_i = \frac{x_0 * R(\theta)}{r_0} * \frac{w_1/2}{\tan(\text{fov}_s/2)} + \frac{w_1}{2} \\
V_i = -\frac{y_0 * R(\theta)}{r_0} * \frac{h_1/2}{\tan(\text{fov}_h/2)} + \frac{h_1}{2} \tag{3.12}
\]

### 3.1.6 Fisheye with lens distortion

The same distortion model is used for the fisheye based camera model as the pinhole based camera model.

\[
r_d = R(\theta) \tag{3.13}
\]

\[
\begin{bmatrix}
x_1 \\
y_1
\end{bmatrix} = (1 + k_2 * r_d^2 + k_3 * r_d^4 + k_4 * r_d^6) \begin{bmatrix}
r_d * x_0/r_0 \\
r_d * y_0/r_0
\end{bmatrix} + \begin{bmatrix}
c_x \\
c_y
\end{bmatrix} \tag{3.14}
\]

\[
U_i = x_1 * \frac{w_1/2}{\tan(\text{fov}_s/2)} + \frac{w_1}{2} \\
V_i = -y_1 * \frac{h_1/2}{\tan(\text{fov}_h/2)} + \frac{h_1}{2} \tag{3.15}
\]

Note that if the perspective "fisheye" formula \( R(\theta) = \tan(\theta) \) is used in Equation 3.13 the fisheye camera model is equivalent to the pinhole mode with lens distortion in Section 3.1.4.

### 3.2 Projector model

The projector is modeled as a camera where the image is the pattern on an imaginary image sensor. The pattern on the imaginary image sensor is the pattern from Figure 2.2 duplicated to a 3x3 matrix with offsets as shown in Figure 3.3.
3.3 Panorama calibration

To calibrate the NIR and RGB camera, an existing in-house Saab Dynamics panorama calibration algorithm was used. Panorama calibration works by mounting the camera on a tripod in such a way that the camera is allowed to rotate around its optical axis 360 degrees. Photos are then taken with at least half image width in overlap, such that all image points are depicted in at least two images. The scene where the images are taken should be static and for best results not have any big areas without structure, (that is no correspondences can be found between two images on a white wall). Assuming that the pixel aspect ratio is known (may it be from the manufacturer or some other calibration method), all the camera parameters must be corrected for the images to fit together seamlessly. For more details on panorama calibration see [Borg, 2007].

3.4 Depth Calibration

The Kinect delivers depth data in a raw format that needs to be converted to depth data in meter. Several formulas to do this have been proposed. In this master thesis will only be checked that the formulas used are reasonable, not trying to calculate the constants in the formulas.

\[
d = \frac{1}{c_1 \cdot r + c_2} \tag{3.16}
\]

\[
d = k_3 \tan\left(\frac{r}{k_1 + k_2}\right) \tag{3.17}
\]

Two formulas are compared: Equation 3.16 from [Burrus] with original values \(c_1 = -0.0030711016\) and \(c_2 = 3.3309495161\). and Equation 3.17 from [Magnenat, 2010] with original values \(k_1 = 1.1863\), \(k_2 = 2842.5\) and \(k_3 = 0.1236\). Based on results from Figure 3.4 the most interesting distance interval to do comparisons in is where the formulas differ most which is in comparative long distance for the Kinect.
Figure 3.4: Difference between the distance conversations formulas 3.16 and 3.17, both with original constants. Note that the distance formulas start to divert significant on longer distances.

Figure 3.5: Comparing the two distance conversions formulas with values measured with a tape measurement for the interval of 7 to 10 meters. The 3.16 formula and 3.17 formula are designated "div formula" and "tan formula" respectively.
From Figure 3.5 it is clear that the formula and constants from [Magnenat, 2010] fits the manual tape measurements much better than the formulas and constants proposed by [Burrus]. The manual tape measurements are accurate to half a centimeter. Note that this conclusion is only valid if the proposed constants are used.

### 3.5 Depth to world coordinates

The Kinect depth data is given in the form of distance from Kinect in ax-direction, which means that if depth images are taken with the Kinect pointing perpendicular to a flat surface ideally, all the depth values will be the same. To transform depth data to world coordinates the method from [Jan Smisek, 2011] is used. This is done by adding an offset to the depth image pixel coordinates to get to NIR image pixel coordinates. The offset used is retrieved from [Jan Smisek, 2011]. The coordinates in the NIR image are used in the camera model for the NIR camera to project the image coordinates onto a plane at the distance specified by the depth data in that particular pixel.

### 3.6 Projector calibration

To understand which camera model to use to model the projector, the analysis is starts with looking at the NIR pattern that is projected. In Figure 3.6 it can be noted that the 9 bright dots are not in a rectangular pattern. The whole pattern is not visible therefore it is hard to estimate an accurate camera model. The projector camera model can be estimated as a pinhole camera with radial distortion and chip offset. The radial distortion being a polynomial function, see Equation 3.7 can be made to fit this model, but this approach is noise sensitive. To be able to see more of the NIR pattern, a second Kinect was used according to the setup in Figure 3.7. Kinect 1 is used only as a NIR camera mounted on a tripod in so that it can be rotated around the optical center of the NIR camera. Five images were taken and warped to a plane representing the wall, the result of which can be seen in Figure 3.8. In Figure 3.8 it can be seen that the projector pattern continu outside of the 3x3 pattern matrix with lower light intensity than inside the 3x3 pattern matrix. The bright dots at the center of the sub pattern have been marked with $p_1 - p_{10}$. $p_{10}$ is outside the 3x3 pattern matrix. The world coordinates of $p_1 - p_{10}$ were calculated in relation to the Kinect 2 projecting the pattern. The $y_c, z_c$ coordinates of $p_1$ were subtracted from $p_1 - p_{10}$. Having done this the five different fisheye formulas $r_p = R(\theta(p))$ was applied to the ten coordinates. Based on the assumption that the all sub patterns are rectangular, of equal size and lie side by side, the following relations are established as desired values:

\[
\begin{align*}
x_r &= \frac{r_{p4} + r_{p5}}{2} \\
y_r &= \frac{r_{p2} + r_{p3}}{2} \\
r_{p6-p9} &= \sqrt{x_r^2 + y_r^2} \\
r_{p10} &= 2x_r
\end{align*}
\] (3.18)
Figure 3.6: Image of Kinect pattern on a flat surface 1.27 m from the Kinect. The 9 bright dots have been marked.
Based on the results in Figure 3.9 it is clear that the orthographic fisheye projection is the projection that comes closest to satisfying the Equation 3.18, especially if the result for $r_{p10}$ is considered. This is no surprise, as single-slit diffraction has maximum at $a \sin(\theta) = n \lambda$ where $a$ is the distance between the slits, $n$ is an integer and $\lambda$ is the wavelength. In Figure 3.10 the result of projecting the image from Figure 3.8 with the orthographic fisheye function can be seen, resulting in a rectangular pattern. After establishing what camera model to use the camera parameters needed to be estimated more precisely. This was done by warping the image from the NIR camera (taken of a flat surface) by a flat surface to the projector. If the models for the camera, projector and plane were perfect, the warped NIR image and projector pattern would match without displacements. The common area between the two images are divided between several sub areas (9x9 for example, where the joints in the 3x3 projector pattern should match joint in the image sub area division). Using area-correlation, the program calculates the displacement between the sub-images. A Newton-Raphson method is used to minimize the displacement between all the sub images, by changing the parameters in projector camera model, pattern displacement and the plane. Then the NIR image is warped again to the projector camera model using the new model and the process is iterated. This is repeated until no significant improvement is achieved by each new optimization.
Figure 3.8: Panorama of Kinect pattern on a flat surface 1.27 m from the Kinect. The white box is what the NIR camera of the Kinect projecting the pattern sees, Figure 3.6. The panorama-image is taken by the first Kinect NIR camera placed below and in front of the projecting Kinect.
Figure 3.9: Results of different fisheye functions, the blue dot are the measured value after transformation and the red circle the ideal grid point result based on formula 3.18. Distance unit [\*] is such that $x_r = 1$. 

---
Figure 3.10: Same panorama as in Figure 3.8, this time seen from the projector position with an orthographic fisheye projection.
3.7 Projector calibration against external camera

To be able to use an external NIR DSLR camera to calibrate the projector, the position and rotation of the camera relative to the projector has to be known. This is achieved by first calibrating the camera using panorama calibration. Then the relative position between the camera and the projector is manually measured.

The goal with image rectification is to simplify the problem of finding correspondences between images. Before image rectification is done, it is generally necessary to search in two dimensions to find correlation between images. After the image has been rectified it is only necessary to search in one dimension. The image rectification is accomplished by projecting the images onto a plane. The plane is chosen in a way that it is parallel to the baseline between the cameras, see Figure 3.11.

As it is hard to exactly measure the position of the camera, the manual measurements are only used as initial values to an image-based calibration method. To determine both camera rotation and camera location a three-dimensional scene is needed. The image from the NIR DSLR camera and the projector pattern are rectified, and an initial value of disparity between the images are calculated based on the depth image from the Kinect. The disparity between the images is used to offset the projector pattern. Then the pattern and NIR images are divided into sub-images.

Using area-correlation the program calculates the displacement between the sub-images. The displacement between all the sub-images is used in a multi-dimensional Newton-Raphson method that tries to minimize the displacement distances by changing the position and rotation of the NIR DSLR. The cost function that the Newton-Raphson tries to minimize is the $\sum (\delta x_n + k \cdot \delta y_n)$ over all subareas $n$, where $\delta x_n$ and $\delta y_n$ are the displacement distance in subarea $n$ in x and y direction respectively, where $k$ is a constant to prioritize the minimization in the offset in y direction. This is because error in y direction is only related to error in camera and projector positions. Offsets in the x direction depend on the distance to the reflective object which is only known from the relatively noisy Kinect depth data. The value used for $k$ in this calibration was set to 10.

After the NIR DSLR camera position has been established it can be used in exactly the same way as the Kinect NIR camera for calibrating the projector pattern.

3.7.1 Vertical line pattern

In the depth image from the Kinect, there are 0, 1, 3, 4 or 5 vertical lines. These lines are created by the raw value systematically being one value higher on one side of the line and lower on the other side of the line. These lines have equal distances from each other plus minus one pixel. It is easy to see the vertical lines in depth images of flat surfaces, but any depth image contains these disturbances. This can be seen by taking the difference between neighboring pixels horizontally and showing the result the interval -1 to 1. The number of lines were found to change between consecutive images without the scene changing, this can be
Figure 3.11: Figure of a rectification plane parallel to the baseline between camera 1 and 2.

seen in Figure 3.12. In Figure 3.13 consecutive images are subtracted from each other. Between image four and image five (image four and five are shown in Figure 3.12) there are vertical line shift and as can be seen in Figure 3.13. There are clear patterns in the differential images that coincide with where the line shifted. Between images where no line shifted, the differential images appear to be random noise.

3.7.2 NIR camera distortion impact on depth

By looking at a depth image of a flat surface it can be seen that the left hand side corners are too far away and that the right hand were corners is too close to the Kinect. This is consistent with the radial distortion in the NIR camera not fully being compensated for in the Kinect internal algorithm.

3.7.3 Projector pattern impact on depth measurements

No visible error can be seen in the joints in the 3x3 diffraction matrix.
Figure 3.12: Consecutive depth images from the Kinect taken from the same position. Note that there are three vertical lines in image (a) and five vertical lines in image (b). The color scale used to visualize this was: black: "no data", dark gray: left pixel > right pixel, gray: left pixel = right pixel and white: left pixel < right pixel.
Figure 3.13: The difference between consecutive depth images that are taken from the same position are shown. Note that the difference between image 3 and image 4 is much smaller than the difference is between image 4 and image 5.
The 3D model is built from depth data captured by a Kinect mounted on a tripod and a color image captured by a DSLR camera. The DSLR is used instead of the Kinect RGB camera due to higher image quality. The Kinect and the DSLR are mounted on an aluminum profile. The aluminum profile in its turn is mounted on a tripod to allow the DSLR to be rotated around its optical center. This setup allows the rotation angle to be estimated using only the image from the DSLR camera. In Figure 4.1 the camera rig is depicted. The NIR DSLR camera is a Nikon D200 with the IR blocking filter removed and a visual light blocking filter placed in front of the camera lens. The DSLR camera is a standard Nikon D300. The Kinect is visible through the hole in the aluminum beam used to hold the Kinect and cameras rigidly together.
Figure 4.1: Image of the rig used to capture the images.
This chapter describes how the validation of the 3D models was achieved. This was done by making a model of two walls and the corner in a room. The model is estimated by using the Kinect for depth measurement and the DSLR for the RGB image. Finally, using data from the model, the angle between the walls is calculated and compared to the real corner. This gives an estimate of the angle accuracy in the model. The robustness of the test setup is also validated by rotating the camera and measuring the variance for the direction of the wall for different camera angles. Two different test setups have been used to perform the test:

1. One image including both walls has been used to model both walls and the angle between the walls. This is done by using only depth data from the Kinect. The result is an estimate of the angle between the walls.

2. A series of images has been taken of one wall, each using a different camera angle towards the wall. The camera has a fixed position and has only been rotated. This method uses both the Kinect depth measurements and the DSLR images. The DSLR images is used to measure the camera rotation. The result is an estimate of the wall direction.

The algorithm that has been used to estimate the flat surfaces, e.g. walls, including handling of data outliers is described in this chapter.

5.1 Representation of a flat surface

With homogeneous representation of a flat surface, the flat surface is represented as a vector of dimension 4 consisting of a normal vector $N$ from the origin to the
flat surface and the distance $L$ to from the origin the flat surface.

$$p = \begin{bmatrix} N \\ -L \end{bmatrix} = \begin{bmatrix} x_N \\ y_N \\ z_N \\ -L \end{bmatrix}$$ (5.1)

in which

$$|N| = \sqrt{x_N^2 + y_N^2 + z_N^2} = 1$$ (5.2)

### 5.2 Algorithm to calculate the flat surface vector

All measurement points from the Kinect depth measurement consist of three elements, $[x_i \ y_i \ z_i]^T$ from point $/(i/).$ In order to create a homogenous representation with a vector of length four, the first three elements are used as the normal vector $N$ and the length $L$ is set to 1.

$$c = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$ (5.3)

To calculate a flat surface, at least three coordinates are needed and they cannot belong to the same line.

$$m = \begin{bmatrix} c_1 & c_2 & \ldots & c_n \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \ldots & x_n \\ x_1 & y_2 & \ldots & y_n \\ z_1 & z_2 & \ldots & z_n \\ 1 & 1 & \ldots & 1 \end{bmatrix}$$ (5.4)

The flat surface $p$ is equal to the eigenvector that corresponds to the smallest eigenvalue of the matrix $m \ast m^T.$ If the two smallest eigenvalues are of the same order of magnitude this indicates that the points in matrix $m$ are close to being on a line [Nordberg, 2011].

### 5.3 RANSAC

RANSAC (RANdom SAmple Consensus) is an iterative algorithm used to find a model that fits data that contains outliers. The algorithm is described in summary below, for a more detailed description see [Hartley and Zisserman, 2004].

1. Randomly select a small number of data points and calculate the model from these points. Small number hens mean the minimum number of data points that gives a unique solution.

2. Count how many data points that can be considered to be inliers.

3. If sufficiently many points are considered to be inliers, go to 4, otherwise
4. The model is re-estimated from all the data considered to be inliers.

RANSAC is a probabilistic method with high probability of reaching a good solution, given good chooses of parameter values. The method will not guarantee to find a solution.

5.4 Angle between flat surfaces on one image

An depth image including two flat surfaces is taken together with the corner between them, where the corner has a known angle. The RANSAC algorithm is used to find a flat surface with the maximum number of inliers. In this test inliers are defined as measurement points that have a maximum distance from the flat surface of 15 mm. When the first surface has been defined, the second surface is being defined by using all the measurement points not being used to define the first surface. The angle between the surfaces is calculated by using the normal vectors \( N \) representing the surfaces. The angle is determined by the equation,

\[
\cos(\theta) = \frac{N_1^T N_2}{||N_1|| ||N_2||}.
\]

This method only uses depth data from the Kinect.

5.5 Angle between flat surfaces from different images

A depth panorama of a flat surface is collected, from each depth frame a flat surface is extracted using RANSAC algorithm and the direction of the surface is calculated. For each step the camera is rotated, the rotation angles between the frames are calculated by using the RGB images. This test method depends on that both the RGB camera and the depth camera have been correctly calibrated. This test gives a better indication compared to test 1, on the precision of entire system.

5.6 Validation test

In the validation of the 3D model two measurement series were taken. For test 1, a series of seven images was taken. The camera rig was rotated in steps of about 15 degrees between different images. For test 2, a series of eight images was taken. The result from test 1, is documented in Table 5.1. In this Table it can be seen that there is no accumulative drift. This can be seen by comparing the data in the diagonal in Table 5.1. It contains the angle between adjacent images. This can be compared with first row containing the angle between images one and all other images. The angle calculated with the method used in this test, always gives a value in the interval 0 to 180 degrees. This means that there are no negative values and that the mean of the values close to 0 will not go towards zero even when the number of measurements approaches infinity and there are no systematic errors. For this reason the mean and standard deviation was only calculated for values close to 90 degrees. The mean and standard deviation for
Table 5.1: Angles between flat surfaces from different images, expressed in degrees. Included in the data are seven images. Image 1 to 5 depict a single wall and image 8 and 9 depict the other orthogonal wall. The comparison in the table is made between image m and image n. The angular difference is presented in the table. The correct value in this table should therefore be 0 and 90 degrees.

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.60</td>
<td>0.53</td>
<td>0.26</td>
<td>1.44</td>
<td>89.75</td>
<td>90.17</td>
</tr>
<tr>
<td>2</td>
<td>0.10</td>
<td>0.86</td>
<td>0.84</td>
<td>89.66</td>
<td>90.07</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.79</td>
<td>0.91</td>
<td>89.59</td>
<td>90.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.71</td>
<td>89.76</td>
<td>90.19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>89.47</td>
<td>89.86</td>
<td></td>
<td>0.87</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2: Table of angle between orthogonal flat surfaces on the same image. Data from image 1 to 8 are used to calculate the angle between the two walls on the image.

<table>
<thead>
<tr>
<th>image</th>
<th>angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>88.94</td>
</tr>
<tr>
<td>2</td>
<td>88.66</td>
</tr>
<tr>
<td>3</td>
<td>89.46</td>
</tr>
<tr>
<td>4</td>
<td>89.26</td>
</tr>
<tr>
<td>5</td>
<td>88.53</td>
</tr>
<tr>
<td>6</td>
<td>88.54</td>
</tr>
<tr>
<td>7</td>
<td>88.64</td>
</tr>
<tr>
<td>8</td>
<td>88.82</td>
</tr>
</tbody>
</table>

The values close to 90 degrees are 89.85 and 0.248 respectively. The correct value is 90 degrees.

In test 2, the angle between the flat surfaces on one image is estimated, the results are presented in Table 5.2. The mean and standard deviation for the eight values in this test are 88.86 and 0.341 respectively. The correct value is 90 degrees.

The suggested method to have a fixed camera position and rotating the camera rig has proven to be an effective way to create a 3D model with a good performance.
6.1 Auto calibration

If you want to do your own calibration of the Kinect, you need to know if it has an automatic internal calibration algorithm. Otherwise the automatic calibration can jeopardize the result. The vertical lines in the depth image that are shifting even when viewing a constant scene give the impression that the Kinect is performing an on-line auto calibration from the scene it is viewing.

6.1.1 Auto calibration test

When testing if the Kinect is using auto calibration the following steps are performed. Start by using two Kinects called A and B. Kinect B is used only as a projector, while Kinect A is collecting the depth data. A and B are placed side by side and the projector on Kinect A is blocked. Kinect A is switched on and then slightly moved until it starts transmitting depth data. Now the projector in Kinect A is unblocked and Kinect B is turned off. After this was done it took a long time before Kinect A started to transmitting depth data again (significantly longer than normal start up time).

6.2 Kinect pattern model

The model for the Kinect pattern works well when calibrated against the NIR images from the Kinect. When the Kinect pattern was calibrated against higher resolution from the images from the NIR DSLR the model proved to be too simplistic.

In Figure 6.1 the white dots show the Kinect projector pattern. The red dots are
calculated positions based on projector pattern, the projector model and depth measured by the Kinect. This figure is done to show the precision of the projector model.

The enlarged part of the image in Figure 6.1 show the consistency between the projector pattern and the calculated pattern. In the enlarged part it can clearly be seen that the calculated red dots are a few pixels from the white dots in the horizontal direction. The horizontal positions of the red dots are independent of depths and only depends on the projector model.

The image in 6.1 was not used for the projector calibration procedure.

Images used for projector calibration show similar results as in fig 6.1, with regards to errors in the depth independent direction. This is an indication of to few parameters in the projector model.

The projector pattern model can be extended to an affine transformation instead of a translation of the sub-patterns to provided the extra parameters needed. This has not been further considered in this thesis.

The vertical differences between the white dots and the reds dots in Figure 6.1 was expected due to limited resolution and inaccuracy in the Kinect depth data.

### 6.3 3D model results

In Figure 6.2, Figure 6.3 and Figure 6.4 you can see illustrations of the classroom Algoritmen at Linköping’s University. The illustrations are all build from a 3D model using depth data from the Kinect and the RGB information from a DSLR camera. The black areas in the figures are areas not visible from the position of the Kinect.

In Figure 6.3 the black circle is a result of the object being closer then the near fielded distance for the Kinect. The model is being build out of 18 images taken from one position for the tripod, and the camera rig is rotated 360 degrees in azimuth. The 3D model is depicted from above, from this perspective it can be seen that the walls are relativity straight.

In Figure 6.4 the same 3D model is depicted from the side.
Figure 6.1: Rectified image from NIR DSLR camera with artificial projector pattern in red superimposed on image. The artificial projector pattern is calculated from the Kinect depth image.
Figure 6.2: A 3D model from one depth image. The 3D model is depicted from a slightly different angle than where it was captured from.
Figure 6.3: A 3D model from 18 images viewed from above. In the middle of the room, where the panorama was taken from, there is a black circle.

Figure 6.4: A side view of the 3D model from 18 depth images.
In this thesis, the Kinects projector pattern has been modeled to the extent that it can be used for stereo measurements. To take full advantage off a higher resolution camera for stereo calculation the model needs to be further improved. An extended model that can benefit from a higher resolution camera has been proposed in this thesis.

The Kinect depth camera has been used together with a high performance DSLR camera. The depth measurement from the Kinect was used together with RGB image from the DSLR camera in the illustration in 3D model results.

Investigation of consecutive images from the same scene shows a systematic change in distance for a vertical segment. This together with the experiment by switching projector between two Kinects, while using only one for detection shows that the Kinect has an online calibration. This makes it difficult to improve the depth measurement by having an individual calibration to a specific Kinect, as the in-built calibration make dynamic adaptations.

### 7.1 Topics for future studies

The following topics for future studies are proposed.

#### 7.1.1 Combined Stereo and structured light sensor

For indoor mapping stereo cameras and structure light sensor can complement each other by stereo giving precise measurement at edges and thin structures and structure light sensor giving good performance on surfaces with little or no texture. In the case of a repetitive pattern, even a structured light sensor with
low performance can give the stereo camera sensor an interval in which to match the patterns.

7.1.2 Combining multiple depth panoramas into one 3D model

Combining data collected from several locations it is necessary to avoid occlusions. To combined multiple depth panoramas into one 3D model there are two main problems that needs to be solved, relative location between the panoramas and some method of combining several meshes into one.

7.1.3 Vertical line detection and complex Residual error compensation

In the paper [Jan Smisek, 2011] depth images of flat surface are taken and a plane surface is fitted to depth data and the average residual of the depth is compensated for. It would be interesting to see if this can get even better by automatically detecting the vertical lines in the depth data and only using data from depth images with the same number of vertical lines for the fixed-pattern noise compensation.
Bibliography


Upphovsrätt
Detta dokument hålls tillgängligt på Internet — eller dess framtidiga ersättare — under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för icke-kommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnt som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright
The publishers will keep this document online on the Internet — or its possible replacement — for a period of 25 years from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for his/her own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/

© Anton Nordmark