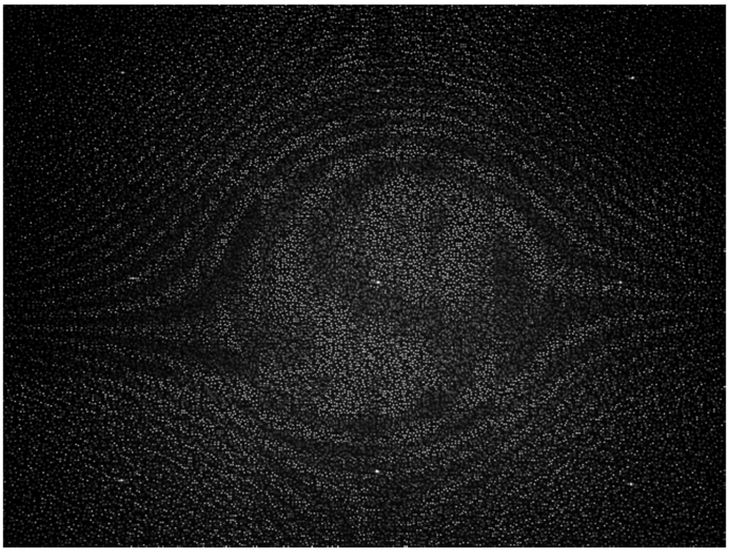
KINECT DEPTH SENSOR EVALUATION FOR COMPUTER VISION APPLICATIONS

Electrical and Computer Engineering
Technical Report ECE-TR-6



AARHUS UNIVERSITY
DEPARTMENT OF ENGINEERING

DATA SHEET

Title: Kinect Depth Sensor Evaluation for Computer Vision Applications
Subtitle: Electrical and Computer Engineering
Series title and no.: Technical report ECE-TR-6

Authors: M.R. Andersen, T. Jensen, P. Lisouski, A.K. Mortensen, M.K. Hansen, T. Gregersen and P. Ahrendt
Department of Engineering – Electrical and Computer Engineering, Aarhus University

Internet version: The report is available in electronic format (pdf) at the Department of Engineering website http://www.eng.au.dk.

Publisher: Aarhus University©
URL: http://www.eng.au.dk

Year of publication: 2012 Pages: 37
Editing completed: February 2012

Abstract: This technical report describes our evaluation of the Kinect depth sensor by Microsoft for Computer Vision applications. The depth sensor is able to return images like an ordinary camera, but instead of color, each pixel value represents the distance to the point. As such, the sensor can be seen as a range- or 3D-camera. We have used the sensor in several different computer vision projects and this document collects our experiences with the sensor. We are only focusing on the depth sensing capabilities of the sensor since this is the real novelty of the product in relation to computer vision. The basic technique of the depth sensor is to emit an infrared light pattern (with an IR laser diode) and calculate depth from the reflection of the light at different positions (using a traditional IR sensitive camera). In this report, we perform an extensive evaluation of the depth sensor and investigate issues such as 3D resolution and precision, structural noise, multi-cam setups and transient response of the sensor. The purpose is to give the reader a well-founded background to choose whether or not the Kinect sensor is applicable to a specific problem.

Keywords: Kinect sensor, Computer Vision, Machine Vision, depth sensor, range camera

Please cite as: M.R. Andersen, T. Jensen, P. Lisouski, A.K. Mortensen, M.K. Hansen, T. Gregersen and P. Ahrendt: Kinect Depth Sensor Evaluation for Computer Vision Applications, 2012. Department of Engineering, Aarhus University. Denmark. 37 pp. - Technical report ECE-TR-6

Cover photo: Image of the infrared pattern that the Kinect depth sensor emits. In this case, on a wooden wall. The image is captured with the infrared camera in the Kinect sensor itself.

ISSN: 2245-2087

KINECT DEPTH SENSOR EVALUATION FOR COMPUTER VISION APPLICATIONS

M.R. Andersen, T. Jensen, P. Lisouski, A.K. Mortensen, M.K. Hansen, T. Gregersen and P. Ahrendt

Aarhus University, Department of Engineering

Abstract

This technical report describes our evaluation of the Kinect depth sensor by Microsoft for Computer Vision applications. The depth sensor is able to return images like an ordinary camera, but instead of color, each pixel value represents the distance to the point. As such, the sensor can be seen as a range- or 3D-camera. We have used the sensor in several different computer vision projects and this document collects our experiences with the sensor. We are only focusing on the depth sensing capabilities of the sensor since this is the real novelty of the product in relation to computer vision. The basic technique of the depth sensor is to emit an infrared light pattern (with an IR laser diode) and calculate depth from the reflection of the light at different positions (using a traditional IR sensitive camera). In this report, we perform an extensive evaluation of the depth sensor and investigate issues such as 3D resolution and precision, structural noise, multi-cam setups and transient response of the sensor. The purpose is to give the reader a well-founded background to choose whether or not the Kinect sensor is applicable to a specific problem.

Chapter 1

Introduction

The Kinect sensor by Microsoft Corp. was introduced to the market in November 2010 as an input device for the Xbox 360 gaming console and was a very successful product with more than 10 million devices sold by March 2011 [1]. The Computer Vision society quickly discovered that the depth sensing technology in the Kinect could be used for other purposes than gaming and at a much lower cost than traditional 3D-cameras (such as time-of-flight based cameras). In June 2011, Microsoft released a software development kit (SDK) for the Kinect, allowing it to be used as a tool for non-commercial products and spurring further interest in the product [2].

The technology behind the Kinect sensor was originally developed by the company PrimeSense which released their version of an SDK to be used with the Kinect as part of the OpenNI organization [3]. The SDK should be independent of devices and, as of now, the company ASUS also produces a sensor with much of the same capabilities as the Kinect sensor (including depth sensing) that works with the OpenNI SDK.

A third approach, and actually the first, to using the Kinect sensor came from the "hacker society" which reverse-engineered the USB-stream of data from the Kinect device (before any SDKs were released). This led to the OpenKinect community that is "working on free, open-source software" on multiple platforms [4].

There has been a lot of interest in the sensor from the Computer Vision society, however, there are not many extensive scientific investigations of the sensor at the moment. In [6], calibration of the sensor is investigated along with issues about multiple sensors with overlapping field-of-views. In [7] the sensor is used as a motion capture system and [10] evaluate the sensor for indoor map building. Careful error analysis of the depth measurements from the Kinect sensor is made in [9].

Our approach is system-oriented and touches on many of the different issues in building real-life Computer Vision systems with the depth sensor. The issues include choice of software framework, depth resolution and precision, spatial resolution and precision, structural noise and multi-cam setups.

Chapter 2

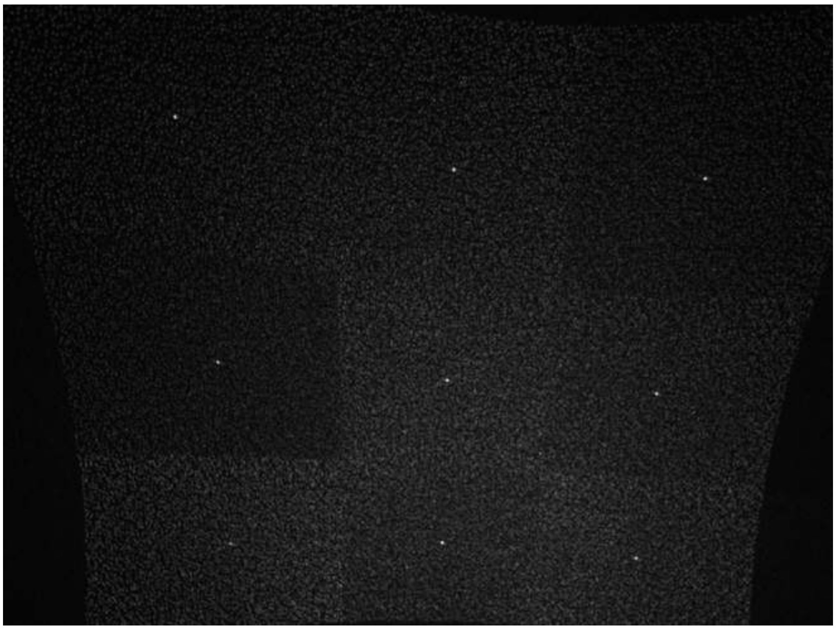
Kinect depth sensor technology

The basic principle behind the Kinect depth sensor is emission of an IR pattern (see figure 2.1a) and the simultaneous image capture of the IR image with a (traditional) CMOS camera that is fitted with an IR-pass filter (figure 2.1b). The image processor of the Kinect uses the relative positions of the dots in the pattern to calculate the depth displacement at each pixel position in the image [8], [9]. It should be noted that the actual depth values are distances from the camera-laser plane rather than distances from the sensor itself (figure 2.3). As such, the depth sensor can be seen as a device that returns (x,y,z)-coordinates of 3D objects.

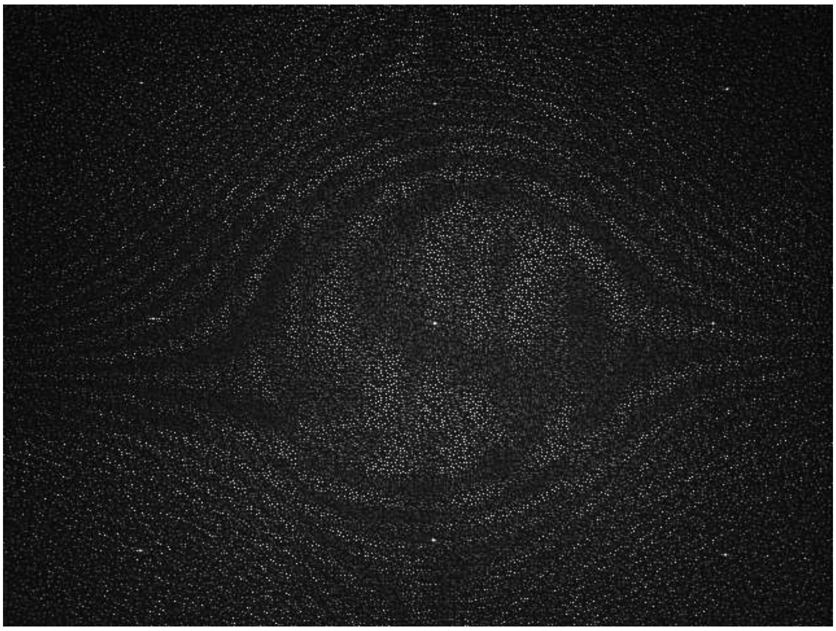
The Kinect hardware specifications have been described in many documents [5], [1]. The main nominal specifications are shown in table 2.1 . These are the specifications that we will evaluate and investigate more carefully in the following.

Property	**Value**
Angular Field-of-View	57° horz., 43° vert.
Framerate	approx. 30 Hz
Nominal spatial range	640 x 480 (VGA)
Nominal spatial resolution (at 2m distance)	3 mm
Nominal depth range	0.8 m - 3.5 m
Nominal depth resolution (at 2m distance)	1 cm
Device connection type	USB (+ external power)

Table 2.1: Kinect hardware specifications



(a) External IR camera



(b) Kinect IR camera

Figure 2.1: The figures show the IR pattern emitted by the Kinect. The pattern has been recorded both with an external IR camera and with the IR camera inside the Kinect. The Kinect IR camera warps the pattern slightly (seen as a swirl in the pattern). Figure 2.2 shows a close-up image of the IR pattern recorded with the external IR camera.

Figure 2.2: The figure shows a close-up of the IR pattern emitted by the Kinect. The image has been recorded with an external IR camera.

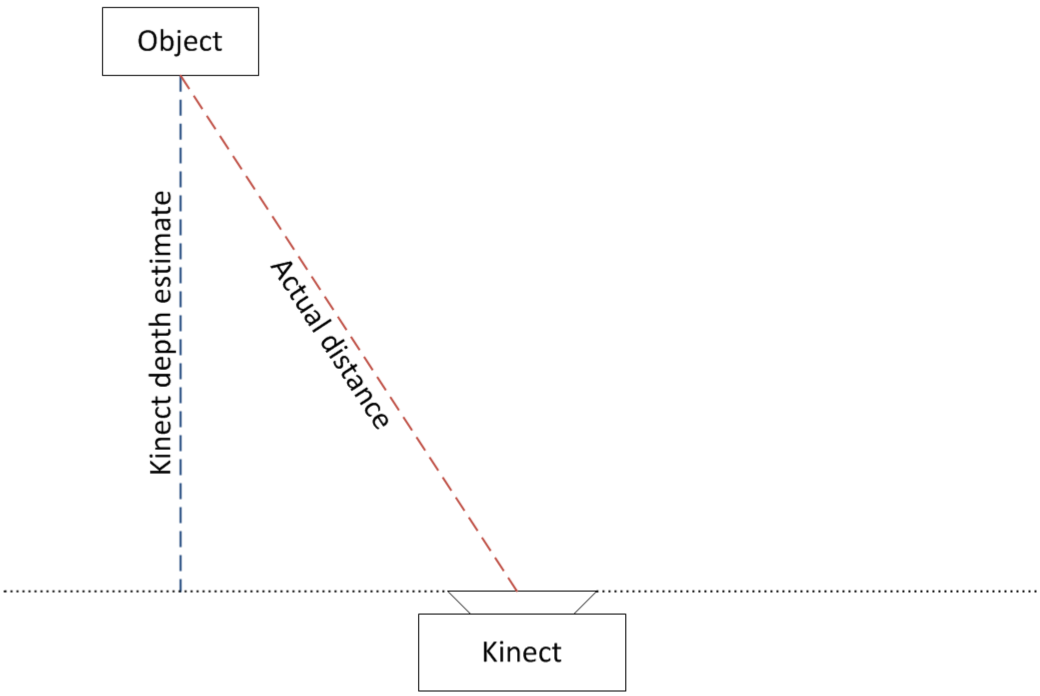


Figure 2.3: The figure describes how the Kinect depth estimates should be interpreted. The depth is an estimate of the distance from the object to the camera-laser plane rather than the actual distance from the object to the sensor.

Chapter 3

Comparison of software frameworks

Currently, three software frameworks are available for the Kinect sensor – Microsoft SDK [2], OpenNI [3] and OpenKinect [4]. Our first investigation relates to the choice between these frameworks. The Microsoft SDK is only available for Windows 7 whereas the other two frameworks are multi-platform and open-source. Another difference is that the Microsoft SDK only delivers depth values up to approximately 4 meters whereas the other two frameworks are up to more than 9 meters. This is likely to be a choice by Microsoft due to low resolution and noise resistance on greater distances, however, our experience is that the depth data can still easily be used beyond 4 meters. The Microsoft SDK is also limited to depths above 0.8 meters whereas the other two frameworks go down to 0.5 meters.

Figure 3.1 illustrates the bit values in relation to actual depth values for each of the three software frameworks. The bit values are calculated in the standard way by converting the bit patterns to unsigned integers. It is seen that the OpenKinect framework delivers uncalibrated data without any linearization. This means that every bit value has a distinct depth value. The other two frameworks instead linearize the bit values such that they represent millimeters exactly. This is obviously easier to use in practice, but also implies that not all bit values are possible. Apparently, the linearization simply corresponds to a look-up table conversion.

Figure 3.2 shows the depth resolutions of the three frameworks. Note that the depth resolution is the smallest possible distance between bit values. It is not surprising that the Microsoft and OpenNI frameworks give approximately

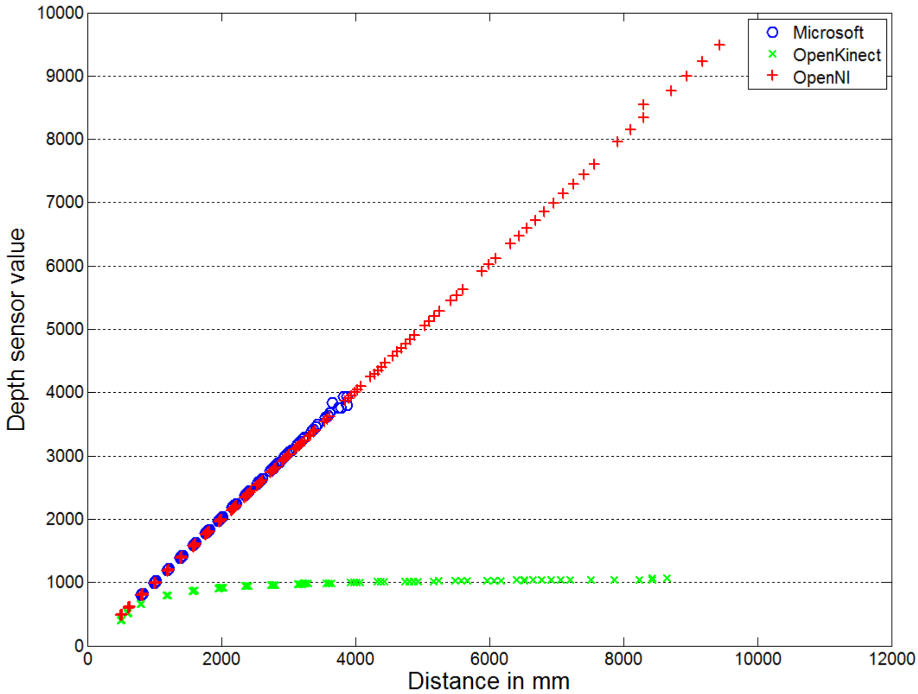


Figure 3.1: The figure shows the bit values in relation to actual depth values for each of the three software frameworks.

similar results since they use the same hardware and are both linearized. The resolution of the two frameworks translate directly to the smallest possible depth difference the Kinect can meassure at a certain distance. E.g. at 6 m the smallest depth difference meassurable is approximately 0.1 m. The most significant difference between the two frameworks is the limited range of the Microsoft SDK. The OpenKinect framework, on the other hand, is not linearized and therefore always has a resolution of 1 bit. As the depth value is not linearized this does not mean that the OpenKinect can meassure smaller depth differences.

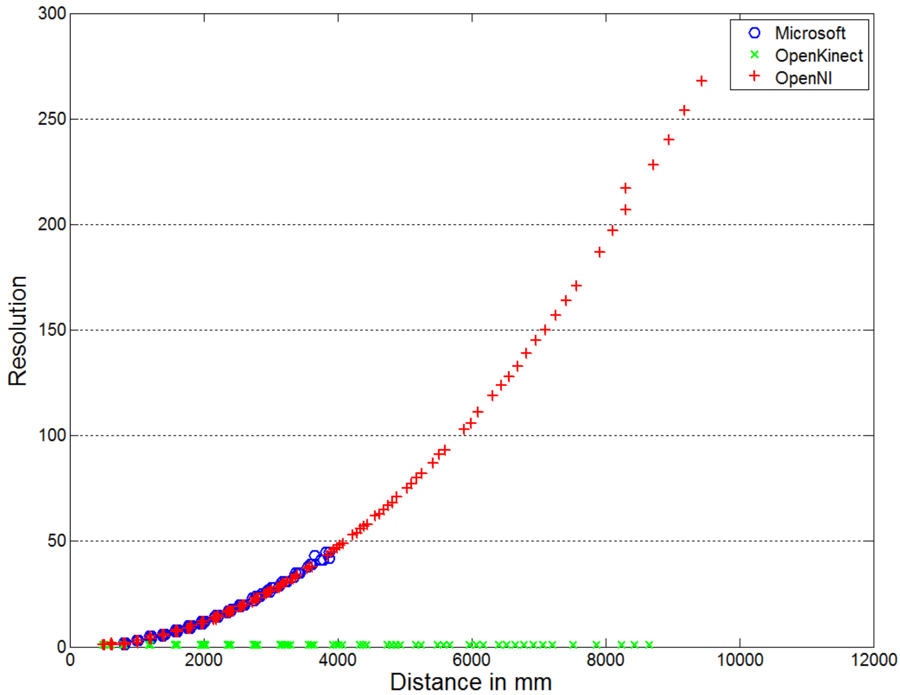


Figure 3.2: The figure shows the depth resolutions of the three frameworks.

The figures 3.1 and 3.2 both come from an initial experiment where a large cardboard box was positioned at different distances from the Kinect sensor and an area of the cardboard box was used to correlate bit values to actual measured depths as well as measuring depth resolutions. The depth resolutions were simply found as the smallest change in depth value in the area on the cardboard box.

After evaluating all three frameworks, we decided to use only the OpenNI framework for the reasons given in the previous. Therefore, the rest of the report and experiments only use the OpenNI framework which is examined in more detail.

Chapter 4

Properties of the sensor

The purpose of this section is to describe the properties of the Kinect depth sensor. A series of experiments constitutes the basis for the description of the resolution, accuracy and precision properties of depth as well as spatial information from the sensor.

All experiments in this chapter have been done using the OpenNI software framework.

4.1 Linearity

To investigate the linearity of the sensor, the sensor is pointed perpendicular towards a planar surface at different distances. For each distance the depth estimates are averaged over a small area at the center of the image and compared to the actual distance. The depth estimates are very close to linear within the measuring range of the sensor. This is shown in figure 4.1, which shows the averaged depth value as a function of distance. The measuring range is approximately 0.5 m to 9.7m. If the limits of the measuring range are exceeded, the depth estimate is set to 0. This indicates that the depth sensor is not able to make a depth estimate. The raw measurements provided by the sensor are non-linear, but the data are linearized in the OpenNI software as described earlier. For more information about the non-linearities of the raw data, see e.g. [11].

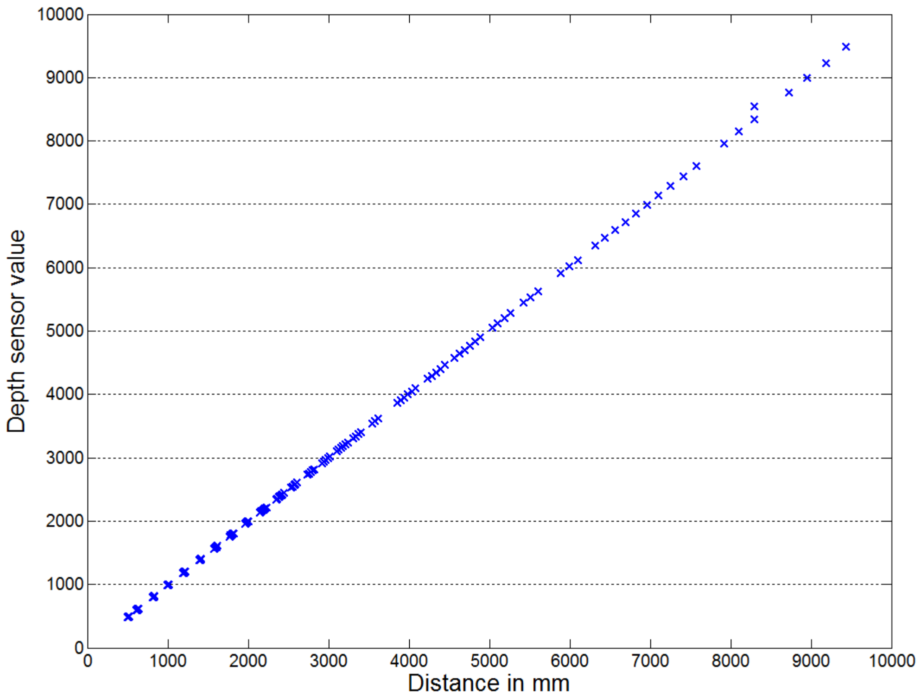


Figure 4.1: The figure shows the depth value from the sensor averaged over a small area as a function of distance. The depth estimates are very close to linear within the measuring range of the sensor.

4.2 Depth Resolution

The experiment for quantifying the resolution is described in the following. In this setup, the depth sensor is pointed perpendicular towards a planar surface and a depth frame is recorded. This is repeated at different distances to the planar surface. For each distance all unique depth values in the recorded frame are extracted and sorted accordingly. Since the real world is continuous, ideally all values between the smallest and largest depth value in a given frame should be present in the sorted depth data. However, since we are dealing with discrete data we are assuming that every possible discrete value between the smallest and largest value is present. The smallest difference between two adjacent values in the sorted list is perceived as the resolution for the given distance.

The result is shown in figure 4.2, which clearly shows that the resolution is getting coarser as the distance increases. Therefore it is necessary to consider needs for range and resolution before using the Kinect sensor for a given computer vision application. The given resolution can be adequate for e.g. navigating a robot through large obstacles, but it might be insufficient if you are trying to pick up chocolate buttons off a production line.

4.3 Depth accuracy and precision

Even if the sensor setup is stationary, if one observes a given depth pixel through time, one will observe that the value of the pixel will fluctuate between a small number of bit levels. To illustrate this phenomenon, we have pointed the Kinect sensor towards a planar surface and recorded the value of a fixed pixel through 1000 frames with a sample rate of 15 frames per second. The actual measured distance from the surface to the Kinect sensor is approximately 2m. Figure 4.3 shows a normalized histogram of the bit values at a specific position. The value of the pixel is fluctuating between four different levels ranging from 2004mm to 2040mm. That is, the precision is roughly 40mm at a distance of 2m.

The spacing between the bins in figure 4.3 suggests that the resolution of the sensor at a distance of 2 meters is approximately 12mm, which is consistent with figure 4.2. However, it is important to emphasize the fact that not all pixels are fluctuating between exactly four different levels. To illustrate this, we have measured the number of levels for all 640x480 pixels

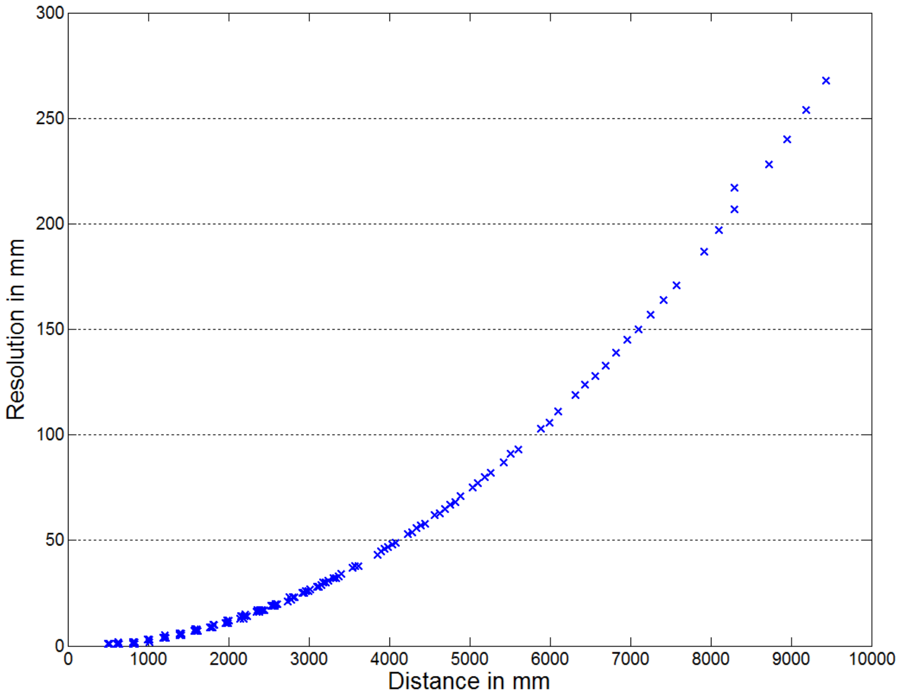


Figure 4.2: The figure shows the depth resolution as a function of distance. Clearly the resolution is getting coarser as the distance increases.

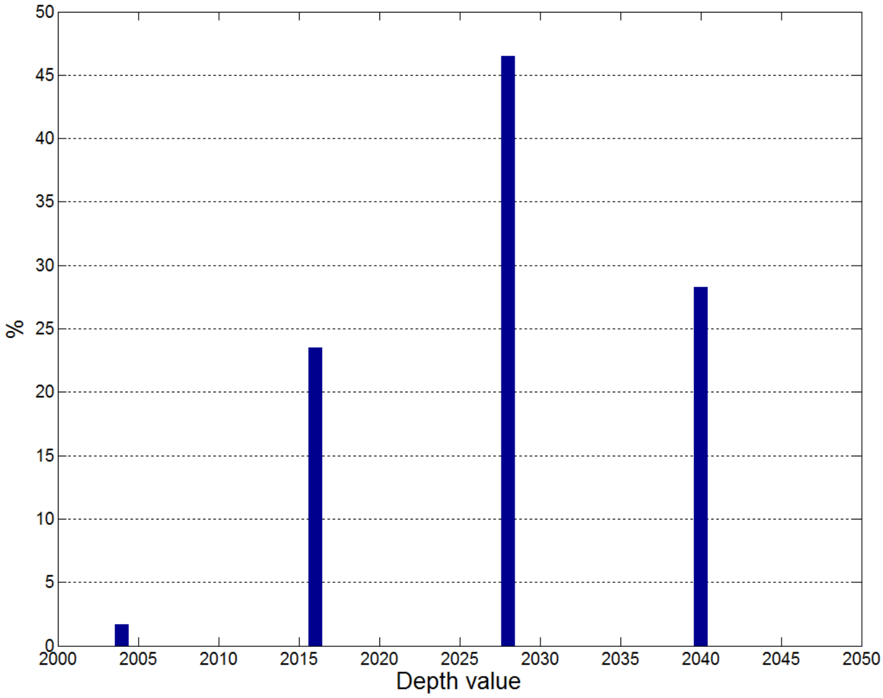


Figure 4.3: The figure shows a histogram for one pixel through time observed in a stationary setup. The value of the pixel has four different levels. The gaps between the bins are caused by the finite resolution.

through a period of time. The result is shown in figure 4.4, which displays a histogram of the number of levels per pixel measured over a period of 10,000 samples with a sample rate of 15 frames per second. Again, the measured object was simply a planar surface.

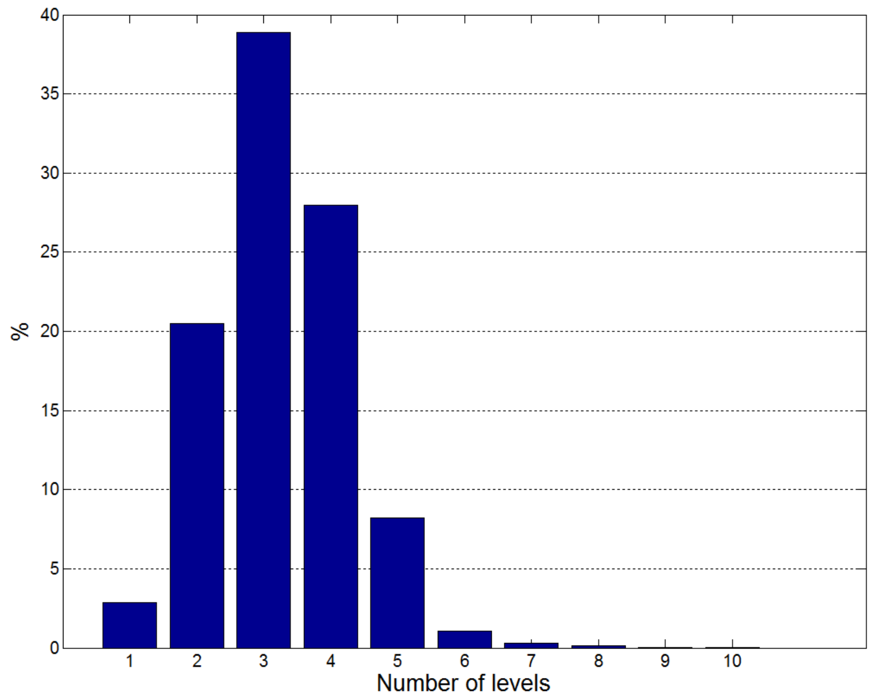


Figure 4.4: The figure shows a histogram of number of levels for each pixel in a frame observed over 10.000 samples. The setup are both stationary for all 10.000 frames. For instance, approximately 2.5% of the pixels have only one level and about 20% of the pixels have two levels.

It appears that with a time span of 10,000 samples roughly 2.5 percent of the pixels have only one level, 20 percent of the pixels have two different levels, 38 percent of the pixel have three different levels and the rest of the pixels have four or more levels. Naturally, the average number of levels per pixel increases with the time span.

To quantify the precision further, the entropy of individual pixels over a period of 1000 frames are calculated. The entropy for each pixel is given by:

$$H(x,y) = -\sum_i p(x,y)_i \ \log_2 p(x,y)_i$$

where $p(x,y)$ is the estimated probability mass function for pixel (x,y). Since

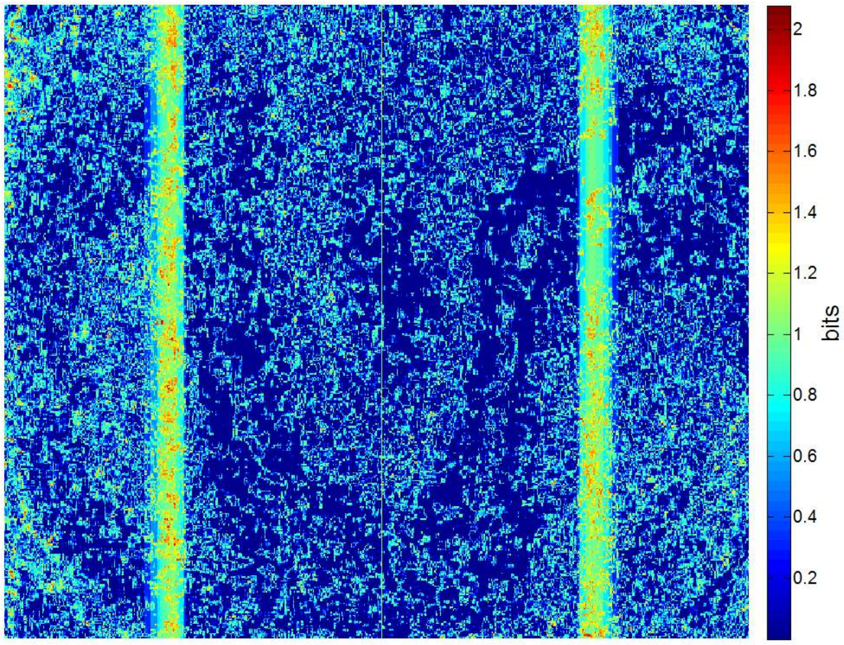
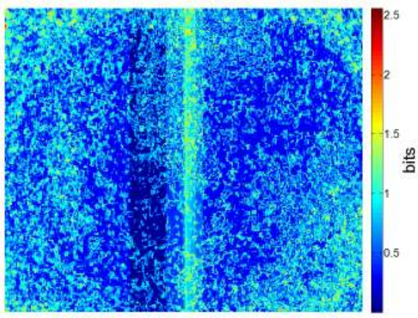
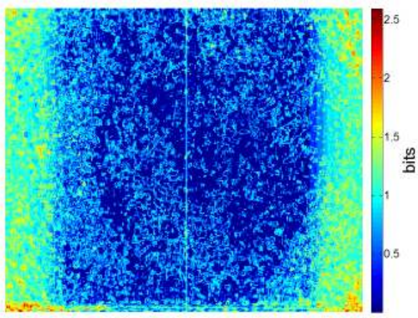


Figure 4.5: The figure shows the entropy of each pixel in a depth frame calculated through 1000 frames at a distance of 1.0m. The narrow bands of increased entropy are caused by some adaptive mechanism. Outside the narrow bands the magnitudes of the entropy are similar.



(a) 1.5m



(b) 2.0m

Figure 4.6: The figures show the entropy of each pixel in a depth frame calculated through 1000 frames at a distance of 1.5m and 2.0m.

entropy can be used as a measure of unpredictability, values as low as possible are most desirable in this situation.

Again, the setup for this experiment is stationary, where the sensor is pointed perpendicular towards a planar surface. This is done for three different distances; 1.0 meter, 1.5 meters and 2.0 meters. The resulting images are shown in figure 4.5 and figure 4.6.

The thing to notice is that there is structure in the images. All three images have vertical bands of different sizes, where the entropy is significantly higher than in the rest of the image. That is, the number of levels for each pixel is larger than in the rest of the image. Our experience suggests that this may be caused by some adaptive mechanism in the way the sensor operates (see section 4.8).

Furthermore, the largest values of entropy for the three images are respectively 2.1, 2.6 and 2.6. Therefore, the distance does not affect the magnitude of the entropy much, but rather the structure.

Entropy is based on probabilities; consequently it does not provide any information regarding the magnitudes of the fluctuations. Thus, the variance of each individual pixel is calculated as well. Figure 4.7 shows the variance of each individual pixel, when the sensor is pointed perpendicular towards a planar surface at a distance of 1.0m. The sample size consists of 1000 frames. In this situation, the variances of all the pixels are similar in magnitude. This is not the case when one or more objects are present in front of the sensor. Figure 4.8a shows the depth image of a slightly more complex scene and figure 4.8b displays the corresponding image of variance. The variances of the pixels along the contours of each physical object, e.g. at the top of the chair, are significantly higher than in the rest of the image. It has been observed that the depth values of these pixels are changing between the distance to the object and the distance to the background. Figure 4.9 shows a plot of the value of a contour pixel as a function of time. As seen in the figure the value of the pixel fluctuates between approximately 1300mm and 2600mm, which correspond to the distance to object and to the background respectively. Notice also that the variances of the pixels near the "shadows" are very large.

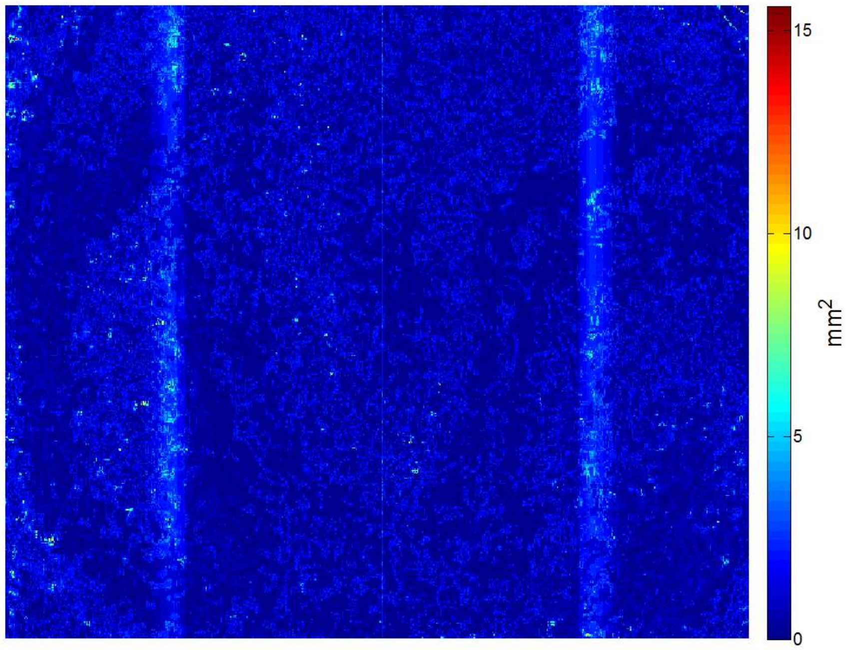
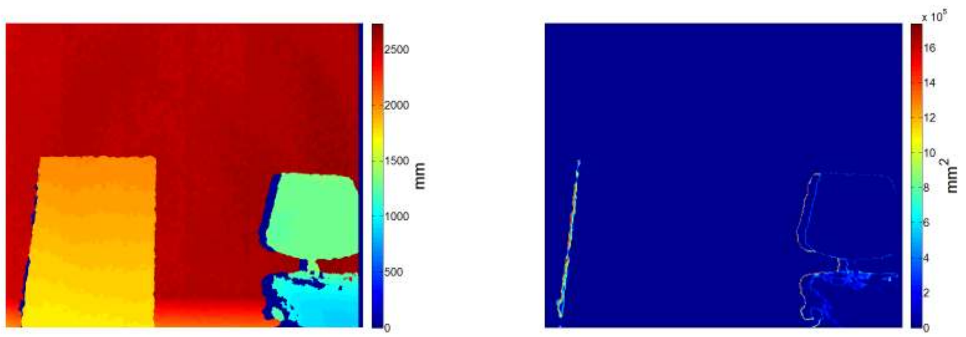


Figure 4.7: The figure shows the variance of each individual pixel in a depth frame calculated through 1000 frames in a stationary setup at a distance of 1.0m.



(a) Depth image.

(b) Variance of individual pixels. Notice the pixels along the contours have significantly higher variance than the pixels in the rest of the image.

Figure 4.8: The figures show a more complex scene

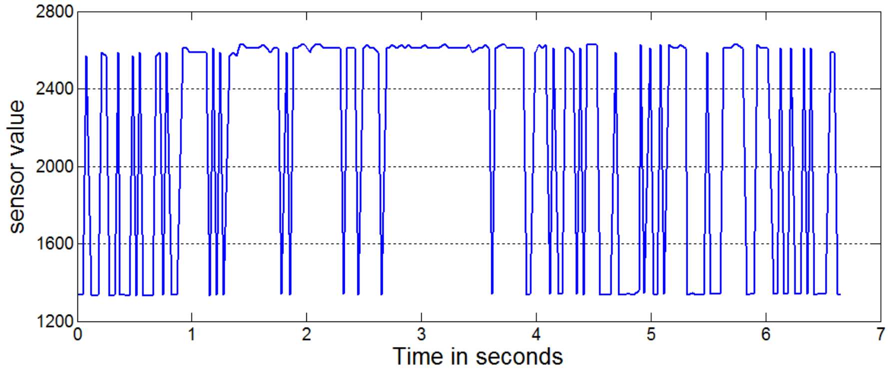


Figure 4.9: The figure shows the value of one pixel with large variance seen over time. The pixel is situated on the top of the contour of the chair shown in figure 4.8. The distance from the Kinect device to the wall is 2.6m and the distance to chair is 1.4m.

4.4 Spatial precision

Figure 4.10 shows a segment of an image showing a top-down view of a box placed 2.2 m from the Kinect sensor. The edges of the physical box are straight lines, but the sensor introduces noise to the edges. These noisy edges are used to analyze the spatial precision of the sensor. In this image, the positions of the edges on the left side of the box are interpreted as the true position plus a noise term and similarly for the top side. For each row the position of the left edge of the box, x_L, is now extracted and the average position, μ_L, is calculated. This is repeated for the top edge of the box. Figure 4.11 shows histograms of the quantities $x_L - \mu_L$ and $x_T - \mu_T$, where x_T and μ_T are the corresponding quantities for the top edge. According to the figures one can expect a spatial precision in magnitude of 4 pixels (roughly 15 mm) in both x and y direction at a distance of 2.2 m.

4.5 Structural noise

The depth estimates in the depth image describe the distance from the point to the sensor plane rather than the actual distance from the point to the sensor. This should result in the same depth estimate over the entire image if the sensor is pointed directly at a planar surface. There are, however, small variations over the image. Figure 4.12 shows an average over 1000 frames from a recording where the sensor was pointed directly at a planar surface. The

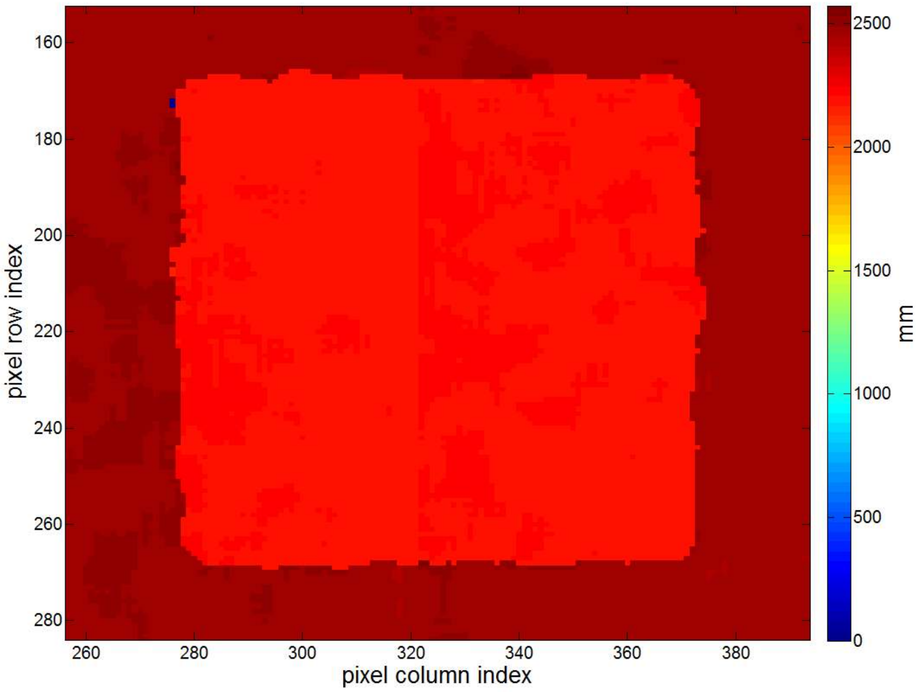
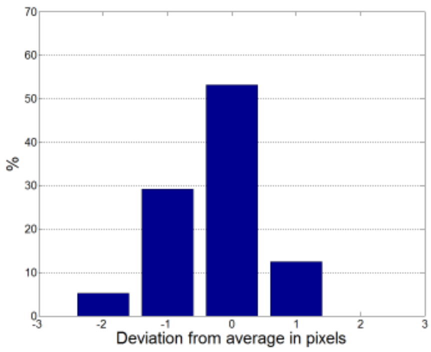
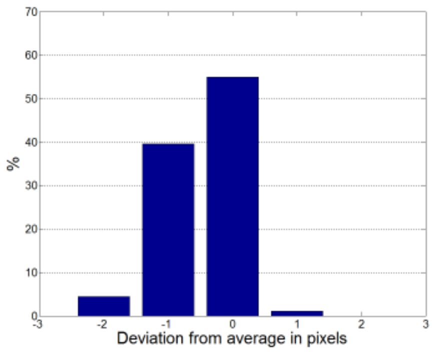


Figure 4.10: The figure shows a segment of a depth image containing a box. The edges of the physical box are straight lines, but the sensor introduces noise to the edges. The noisy edges are used to analyze the spatial precision of the sensor.



(a) Deviation of position of the leftmost edge.



(b) Deviation of position of the topmost edge.

Figure 4.11: The figures show histograms of the deviation of position of the leftmost and topmost edge in figure 4.10. Based on this sample, one can expect a spatial precision of 2 pixel in both the horizontal and the vertical direction.

sensor was located 1.0m from the surface, but as the figure shows the depth estimate varies by roughly 40 mm. This variation is not entirely random and appears like a circular ripple in the depth image.

The same circular ripple continues to appear in the image as the distance from the sensor to the surface is increased. The variation in the depth estimate also increases. This can be seen in figure 4.13. At 1.5m the depth varies roughly 80 mm and at 2.0m this is further increased to roughly 130mm. The increase in variation corresponds to the depth resolution getting coarser at longer distances (see figure 4.2). That is, the variation in bit levels is roughly the same at all three observed distances. The variation might come from the transformation the sensor performs on the depth data so it represents the distance from point to sensor plane. Another possibility is that the sensors spatial resolution is not sufficient to record the IR pattern correctly or that the lens warps the IR image slightly. These are merely speculations though.

4.6 Multi-camera setup

It is possible to cover larger areas with multiple sensors. As the depth image is based on an infrared image of a pattern projected onto the scene by a laser, multiple sensors may interfere with each other. If the pattern projected by

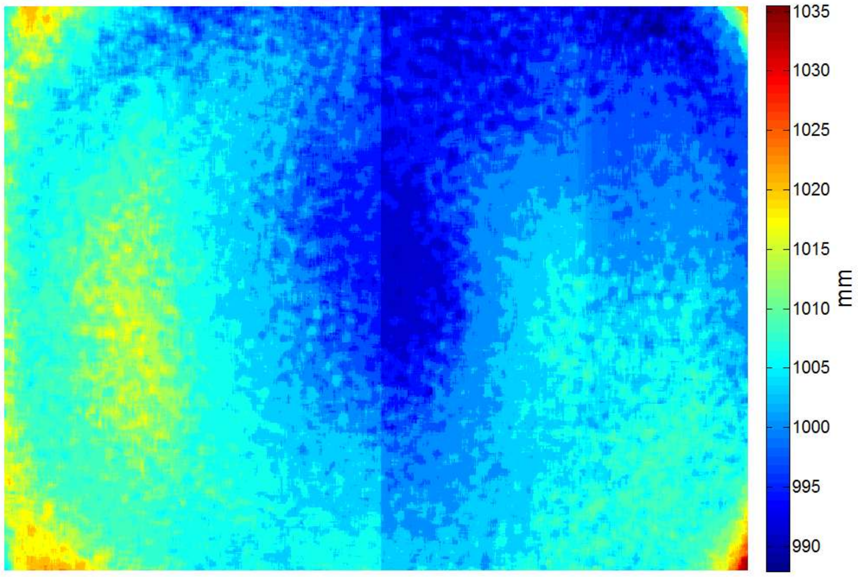
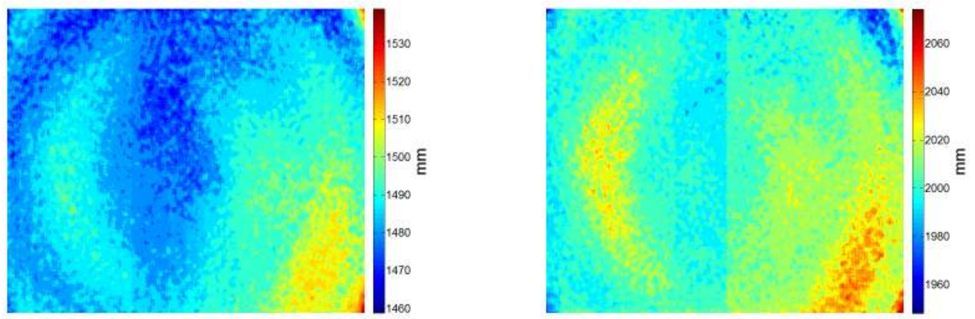


Figure 4.12: The figure shows the depth image of a planar surface at a distance of 1.0m. The pixel values describe the depth estimate from the surface to the sensor plane in millimeter. Ideally the depth estimate should be identical over the entire image, but as it is seen on the image the depth estimate varies by roughly 40mm. This variation appears in the image as a circular ripple.



(a) 1.5m

(b) 2.0m

Figure 4.13: The figures show the depth image of a planar surface at two distances. The circular ripple in depth appears at both distances.

one IR laser overlaps with another, the sensor may not be able to make an estimate of the depth. This shows itself in the depth image as blind spots in the overlapping area where the depth estimates are set to 0. Contrary to what would be expected, the sensor is able to make a correct depth estimate in almost all of the overlapping area. Figure 4.14 shows the percent of the overlapping area which are blind spots as a function of the overlap.

Figure 4.14 was created from an experiment with two depth sensor that were pointed directly at a planar surface. One of the sensors was then moved sideways to increase the overlapping area. A sketch of the setup can be seen in figure 4.15.

The position and size of the blind spots depend on how the two IR patterns of the sensors overlap. An example of a depth image with blind spots can be seen in figure 4.16. Except for the blind spots, the depth estimates in the image are not affected by the overlap. That is, the depth values of the non-blind pixels in the overlapping area appear to be completely unaffected by the interference from the other sensor.

Another problem arises in a situation, where the purpose is to record an object from more than one viewpoint. Figure 4.17 shows a sketch of such a situation, where two Kinect devices are facing each other. The same problems as previously mentioned occur in areas where the laser patterns of the sensors overlap, but in addition to this there is the possibility that the laser pattern from one sensor may "blind" the camera of the other sensor. With a well-placed setup of the sensors, the overlapping areas can be minimized and the

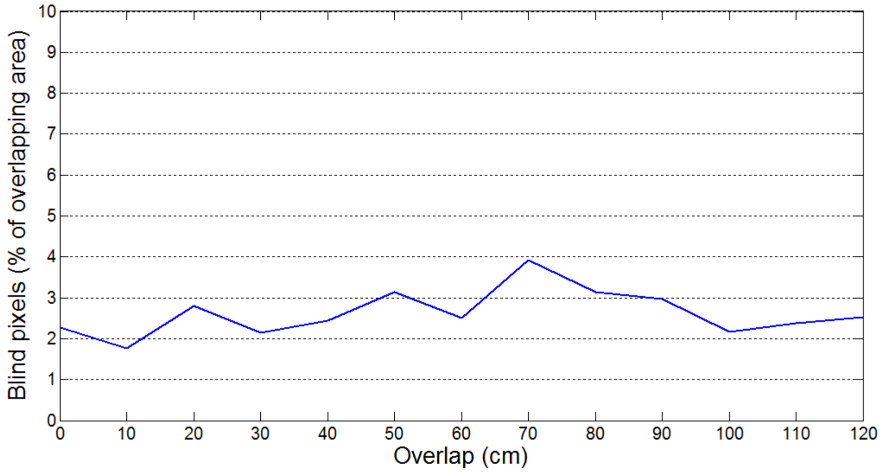


Figure 4.14: The figure shows the percent of blind pixels in the overlapping area. The percentage is roughly constant as the overlap increases.

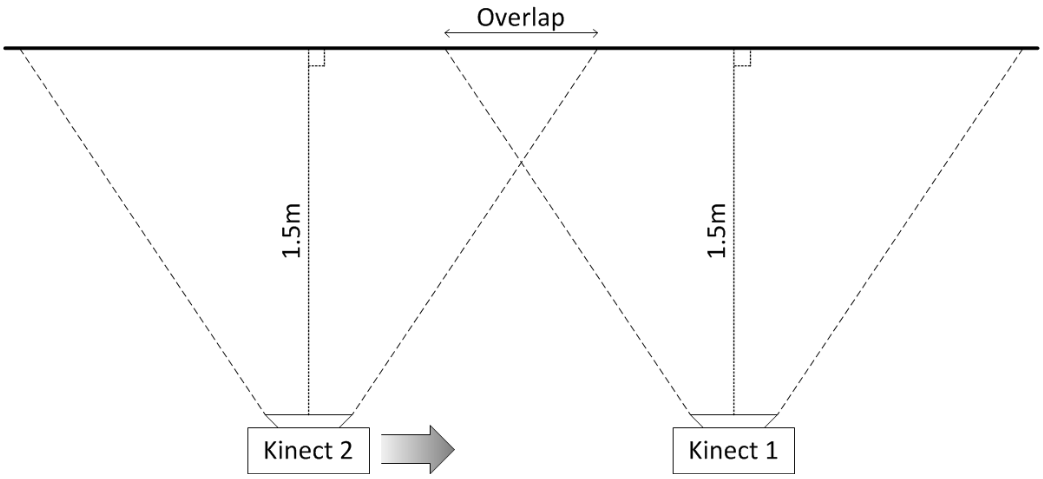


Figure 4.15: The figure shows the setup used to measure the interference caused by overlap. Kinect 1 is stationary while Kinect 2 is moved sideways in steps of 100mm to increase the overlap.

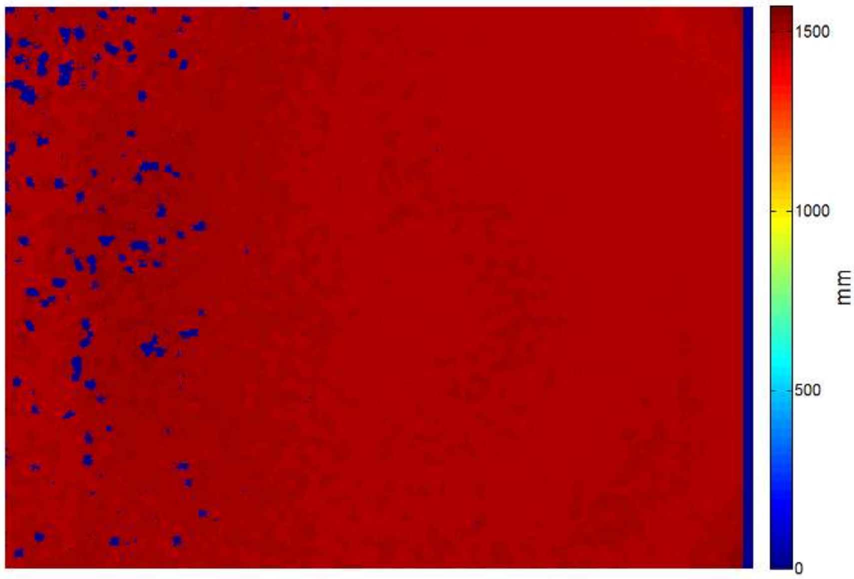


Figure 4.16: The figure shows the depth image from a Kinect pointed directly at a planar surface. The blind spots (0mm depth estimate) are caused by interference from another Kinect.

blinding avoided. As an example, two sensors have been placed such that they face each other to record both sides of a person in a room (see figure 4.17). Figure 4.18 shows the RGB and depth images of the two Kinects.

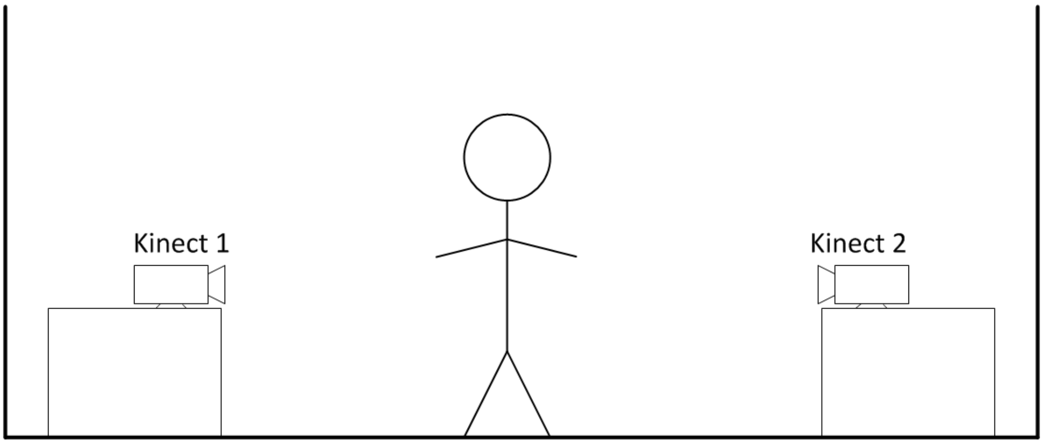


Figure 4.17: The figure shows the setup used for recording both sides of a person.

4.7 General considerations

4.7.1 Lens Distortion

The infrared sensor of the Kinect has been tested for lens distortions. The test did not show any noticeable effects on the images. Figure 4.19a shows a 1m ruler recorded with the infrared sensor of the Kinect. The laser on the Kinect illuminates the area. Figure 4.20a shows the pixel intensity of a line across the rulers centimeter markings in the image. A spike in intensity is seen for the white markers. The width of these spikes does not vary from the center to the edge of the image which indicates that there are no noticeable wide-angle effects. The test has been performed both horizontally and vertically. See figure 4.19b and figure 4.20b for the vertical case.

4.7.2 Calibration

Due to the distance between the IR camera and the RGB camera in the Kinect device, the depth- and RGB-images are not aligned. This can be seen on figure 4.21a, where the depth image is superimposed onto the RGB image. The figure clearly shows the misalignment of the images. The OpenNI

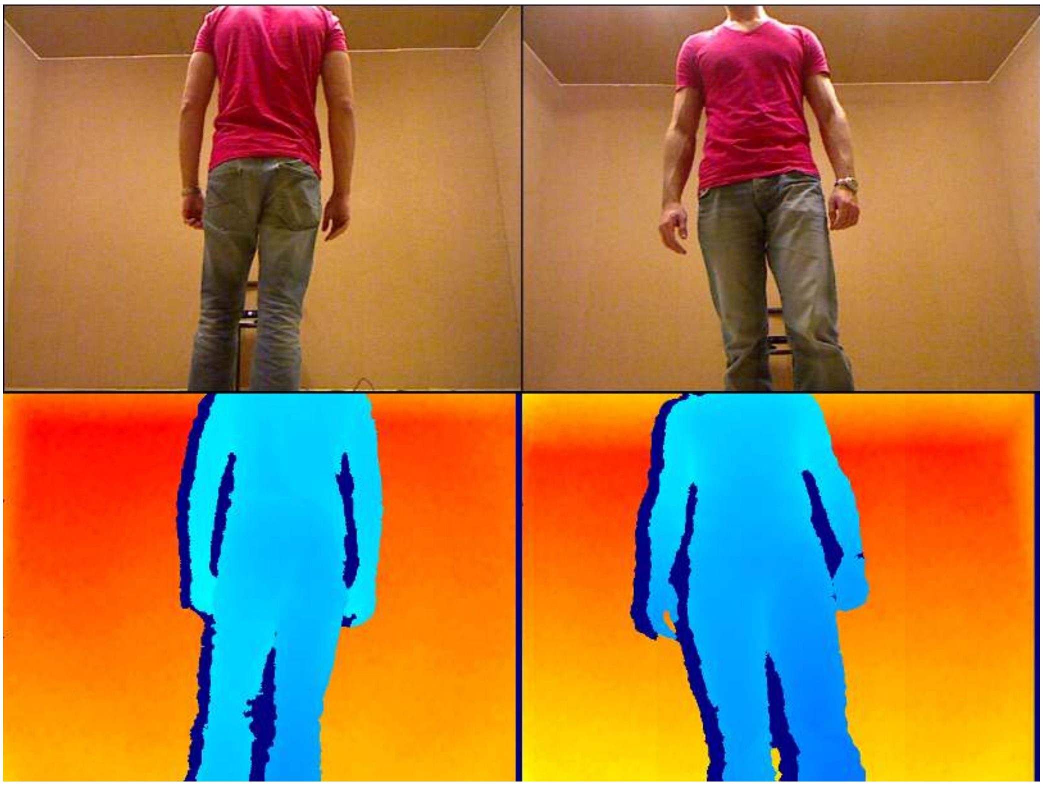
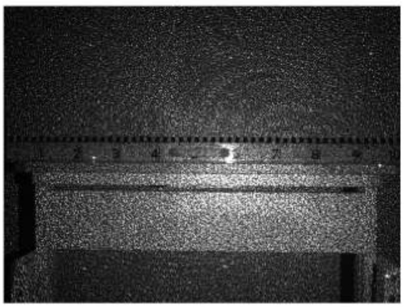
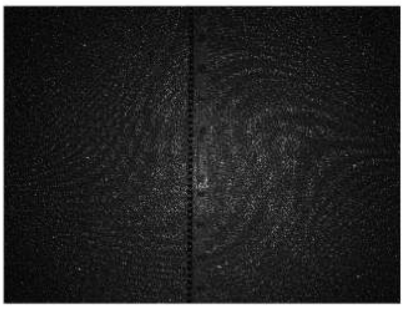


Figure 4.18: The figure shows the RGB images (top) and depth images (bottom) from two Kinects recorded simultaneously. The setup is as shown on figure 4.17.

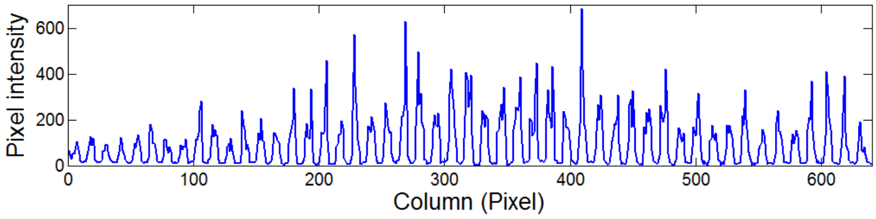


(a) Horizontal

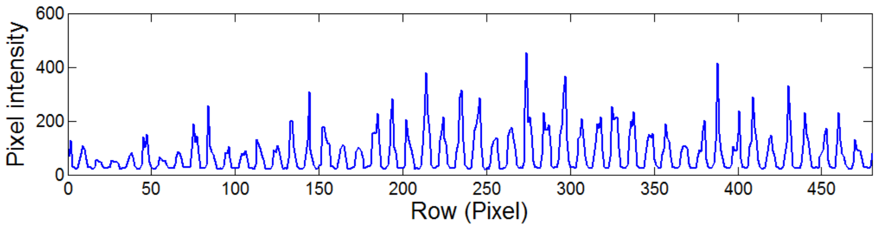


(b) Vertical

Figure 4.19: The figures show images recorded with the Kinect IR sensor. A 1m ruler has been placed horizontally and vertically in the scene to investigate the lens distortion properties of the Kinect IR sensor. The scene is illuminated by the Kinect IR laser.



(a) Horizontal



(b) Vertical

Figure 4.20: The figure shows the pixel intensity of a row/column along the centimeter markings on the horizontal and vertical rulers in figure 4.19. The spikes in the figure correspond to the white markings on the ruler. The widths of the spikes are roughly constant from edge to center of the images which suggests that there is little or no lens distortion (or the image has been calibrated in the Kinect/driver).

framework provides functionality for aligning the two sensor images. Figure 4.21b shows the result of the alignment. The size of the depth image is decreased because the field-of-view of the RGB sensor is larger than the field of view of the depth sensor.

(a) It is easily seen that the two images are misaligned.

(b) The two images are now aligned using the built-in functionality in the OpenNI driver.

Figure 4.21: The figures show an RGB image superimposed with the corresponding depth image. In the first image the depth has not been aligned to the RGB image. In the second the built-in OpenNI function has been used to align depth and RGB.

4.7.3 Shadows in the depth image

Due to the distance between the illuminator and the IR camera in the Kinect device, illuminated objects may cause shadows in the depth image. The shadow in the pattern clearly makes it impossible for the sensor to estimate the depth and therefore the pixels in the shadowed area are set to zero depth. Figure 4.22 explains why shadowing occurs.

Figure 4.23 shows the depth- and RGB-images of a chair. The shadows in the depth image are clearly seen on the leftmost edge of the chair. Shadows will always appear on this side of an object due to the physical positions of the IR camera and the IR laser.

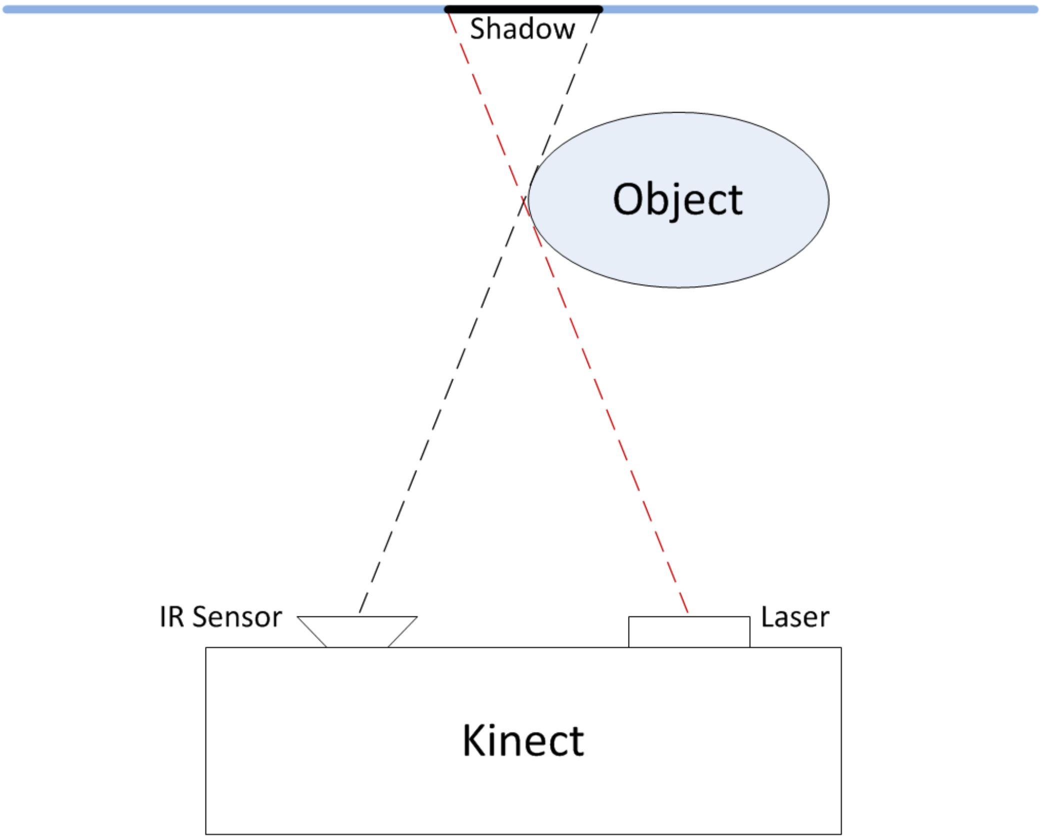
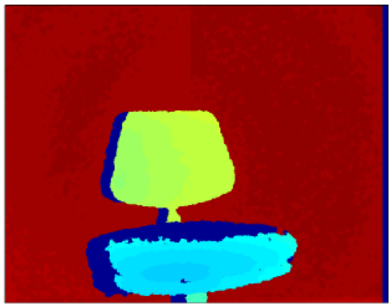


Figure 4.22: The figure shows a sketch of why shadows occur in the depth image. The object in the figure blocks the path of the laser. Since the depth estimate is based on the pattern projected by the laser, it is not possible for the Kinect to estimate distance outside the line of sight of the laser. The IR sensor is positioned to the left of the laser which means that the shadows occur on the left side of an object.

(a) The figure shows the RGB image of a chair used to illustrate the shadows caused by the distance between the laser and the sensor.



(b) The figure shows the depth image of the chair. A shadow occurs on the left edge of the chair (dark blue area with 0 depth). The shadows occur on the left side due to the relative position of the laser and the sensor.

Figure 4.23: The figures show the RGB and depth image of a chair.

4.8 Transients

During experiments we observed that from the time we power on the device, it has a settling time before the pixel values reach their final state. Figure 4.24 shows the pixel value for the first 90 seconds after powering on the device. It is clear from the figure, that the device needs approximately 30 seconds to reach its final value. Equivalently, if the sensor is rotated and quickly pointed towards a different scene, a similar adapting sequence can be observed. This might be important in applications, where the sensor itself is moving e.g.for robot vision applications.

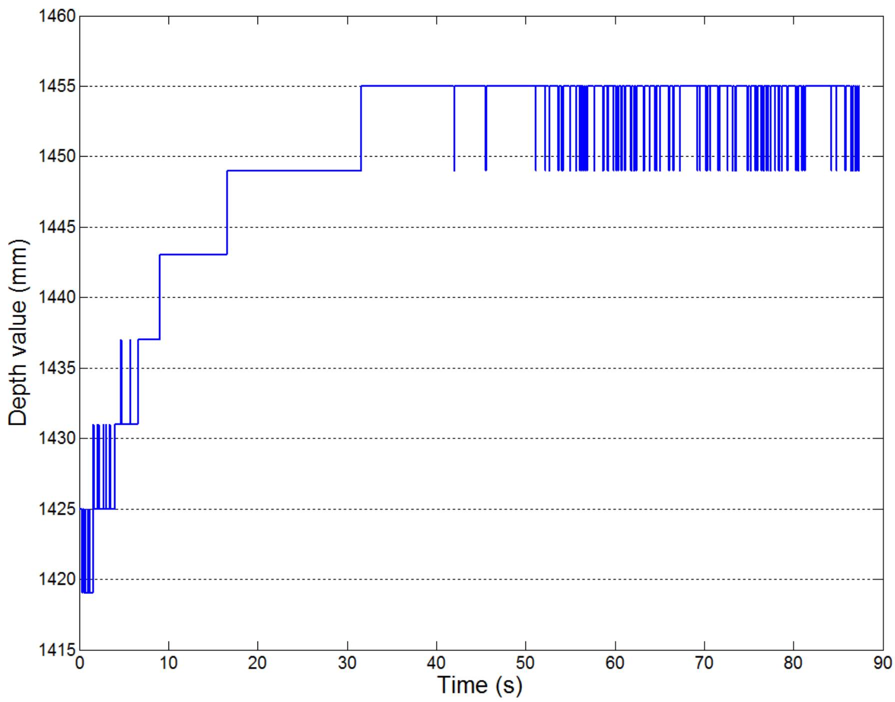


Figure 4.24: The figure shows the value of a pixel as a function of time after powering on the Kinect device. It is clear, that the system is in a transient state before reaching its final state after approximately 30 seconds.

Chapter 5

Conclusion

In this report, we have investigated many of the topics that are important when the Kinect depth sensor (or a similar device with the same reference design) should be used in practical, real-life Computer Vision systems. We have examined software frameworks, depth and spatial resolution, structural noise and many other effects that occur in this complex device.

There are also many other parameters that will influence the sensor in a given situation such as the characteristics of the viewed objects. For instance, there are obvious problems with reflective materials which the sensor is simply not able to determine a distance to. This is clearly understood since the sensor emits IR light which will not be reflected back to the camera. Similarly, diffusely reflective material with very low reflectivity will not reflect enough light for the camera to be able to determine distance to the object. This is obvious when the technology behind the Kinect is considered, however, it still limits the applicability of the sensor in different setups. Most alternative distance sensors will suffer from the same defects.

Another complication is interfering IR light from e.g. the Sun. This clearly limits the usefulness of the sensor in an outdoor setting. From a practical viewpoint, even a strong ray of sunlight may blind the sensor indoor and other sources of household lighting may influence the sensor, although we did not experience many problems in our projects. One solution to the IR-interference problem might be to employ the RGB-images when the depth-sensor fails and vice versa.

Hopefully, this report will give the reader a background to decide whether or not the Kinect depth sensor will be useful for his/her application. We have surely found it useful in many of our projects.

Bibliography

[1] http://en.wikipedia.org/wiki/Kinect.

[2] http://www.microsoft.com.

[3] http://www.openni.org.

[4] http://www.openkinect.org.

[5] http://www.ros.org/wiki/kinect.

[6] Kai Berger, Kai Ruhl, Christian Brümmer, Yannic Schröder, Alexander Scholz, and Marcus Magnor. Markerless motion capture using multiple color-depth sensors. In *Proc. Vision, Modeling and Visualization (VMV) 2011*, pages 317–324, October 2011.

[7] Tilak Dutta. Evaluation of the kinect sensor for 3-d kinematic measurement in the workplace. *Applied Ergonomics*, October 2011.

[8] Barak Freedman. Method and system for object reconstruction. US Patent US 2010/0118123 A1.

[9] K Khoshelham. Accuracy analysis of kinect depth data. *GeoInformation Science*, 38(5/W12):1–6, 2010.

[10] Todor Stoyanov, Athanasia Louloudi, Henrik Andreasson, and Achim J Lilienthal. Comparative evaluation of range sensor accuracy in indoor environments. *Evaluation*, 2011.

[11] Mikkel Viager. Analysis of kinect for mobile robots. Technical report, Department of Electrical Engineering, Technical University of Denmark, march 2011.

M.R. Andersen, T. Jensen, P. Lisouski, A.K. Mortensen, M.K. Hansen, T. Gregersen and P. Ahrendt:
Kinect Depth Sensor Evaluation for Computer Vision Applications, 2012

Department of Engineering
Aarhus University
Edison, Finlandsgade 22
8200 Aarhus N
Denmark

Tel.: +45 4189 3000