8-1-2015

Characterization of Different Microsoft Kinect Sensor Models

Nicholas M. DiFilippo
University of Rhode Island

Musa Jouaneh
University of Rhode Island, jouaneh@uri.edu

Citation/Publisher Attribution

DiFilippo, Nicholas M., and Musa Jouaneh. "Characterization of Different Microsoft Kinect Sensor Models." *IEEE Sensors Journal* 15, 8 (2015): 4554-4564. doi: 10.1109/JSEN.2015.2422611.

Characterization of Different Microsoft Kinect Sensor Models

Keywords

3-D image reconstruction; depth; Kinect accuracy; Kinect for windows; Kinect for Xbox; Kinect sensor; OpenNI

N. DiFilippo and M. Jouaneh, "Characterization of Different Microsoft Kinect Sensor Models", IEEE Sensors Journal, No. 8, pp. 4554-4564, 2015.

Characterization of Different Microsoft Kinect Sensor Models

Nicholas M. DiFilippo and Musa K. Jouaneh, *Senior Member, IEEE*

***Abstract*— This experimental study investigates the performance of three different models of the Microsoft Kinect sensor using the OpenNI driver from Primesense. The accuracy, repeatability, and resolution of the different Kinect models' abilities to determine the distance to a planar target was explored. An ANOVA analysis was performed to determine if the model of the Kinect, the operating temperature, or their interaction were significant factors in the Kinect's ability to determine the distance to the target. Different sized gauge blocks were also used to test how well a Kinect could reconstruct precise objects. Machinist blocks were used to examine how well the Kinect could reconstruct objects setup on an angle and determine the location of the center of a hole. All the Kinect models were able to determine the location of a target with a low standard deviation (< 2 mm). At close distances, the resolutions of all the Kinect models were 1 mm. Through the ANOVA analysis, the best performing Kinect at close distances was the Kinect model 1414, and at farther distances was the Kinect model 1473 .The internal temperature of the Kinect sensor had an effect on the distance reported by the sensor. Using different correction factors, the Kinect was able to determine the volume of a gauge block and the angles machinist blocks were setup at, with under a 10 percent error.**

***Index Terms*— Kinect Sensor, Kinect Accuracy, 3-D Image Reconstruction, Kinect for Xbox, Kinect for Windows, OpenNI, Depth Measurement.**

I. Introduction

THE Microsoft Kinect is a low cost sensor that integrates many components. It is composed of a traditional RGB camera, a depth sensor consisting of an infra-red (IR) camera and projector, a microphone, and a built-in motor. The built-in motor allows the base of the Kinect sensor to tilt [1]. The Kinect sensor was originally designed to be used to control games on the Microsoft Xbox 360 gaming console. Using their hand with the Kinect sensor rather than a controller, players are able to control characters or menus in a game. Instead of being limited to only gaming applications, multiple libraries are available that open up potential applications of the Kinect sensor to fields involving computer vision [2], 3D mapping [3-5] , robotics [6-10], medicine[11-18], human tracking [19,20] , as well as others.

Nicholas M. DiFilippo is a graduate student in the Mechanical Engineering Department at the University of Rhode Island, Kingston, RI 02881 USA (e-mail: ndifilippo@my.uri.edu).

Musa K. Jouaneh is with the Mechanical Engineering Department at the University of Rhode Island, Kingston, RI 02881 USA (e-mail: jouaneh@egr.uri.edu).

Examples of these applications include the work of Shirwalker et al. [9] and Afthoni et al. [10] who have both used the Kinect sensor to obtain gestures that an operator makes in order to control a robotic arm. Alnowami et al. [12] and Tahavori et al. [13, 14] used the Kinect to monitor patients' respiratory breathing patterns while they are receiving external beam radiotherapy. It was found that the Kinect performs quite well and can determine various types of breathing patterns compared to the equipment that is currently used. Ning and Guo [17] showed that the Kinect sensor can be used to assess the spinal loading of a person. Yang et al. [18] assess how the Kinect sensor performs in measuring the postural stability of a person. It was reported that although the Kinect sensor needs to be calibrated by a set of linear equations, it was able to measure a person's standing balance comparable to how standard testing equipment is able to. Obdrzálek et al. [21] used the Kinect sensor to estimate human poses and compared this with other known techniques. They found that with controlled postures, such as standing and exercising arms, the Kinect's performance is comparable with motion capture techniques. However, the Kinect's estimation of general postures can be off by as much as 10 cm and the Kinect skeletal tracking often fails due to self-occlusions of body parts.

The three main libraries that allow programmers access to the Kinect's camera and depth information are OpenKinect, the Microsoft Kinect Software Development Kit (SDK), and OpenNI [1]. The first library available for developers, OpenKinect , was released in November 2010 via the hacker community [22]. PrimeSense, creator of the hardware in the Kinect sensor, released the OpenNI open source SDK in December 2010. OpenNI is a framework that uses the middleware library, NiTE, to enhance the Kinect sensors gesture recognition and tracking abilities. In June 2011, Microsoft released the Microsoft Kinect SDK [23] which allows development of applications using C++, C#, and Visual Basic.

The two types of Kinect sensors available are the Kinect for Xbox 360 and the Kinect for Windows (K4W). The main difference between these sensors is the range at which the sensors can return depth values for using the Microsoft Kinect SDK. While the Kinect for Xbox 360 can only determine the distance of an object between 800 and 4000 mm, the K4W possesses a near mode option allowing it to change this range to see objects between 400 and 3000 mm. Additionally, the Kinect for Xbox 360 has two common models, model 1414 and model 1473. The OpenNI drivers and the Microsoft

Kinect SDK work with all of the sensors however; the OpenKinect libraries currently only support the Kinect model 1414. At the time of this writing, Microsoft has released a new Kinect for Windows v2. This sensor has a full 1080p video, a higher depth fidelity, and can track up to 6 people. However, it is only available on Windows 8 or 8.1 and can only interface with the Microsoft SDK V2[24]. This new Kinect sensor works by using time of flight (ToF) technology [25] where a light source transmits a modulated light signal that travels to an object and is then reflected back to the sensor. The Kinect uses phase detection that measures how long it takes the light to travel to the object and back and can then determine the depth of an object from this time.

The Kinect Sensor works by using a speckle pattern 3-D mapping [4, 26]. This is when the sensor projects an IR speckle pattern on the scene and compares this pattern to a reference pattern. If an object in the scene is farther or closer than the reference plane, the speckles that are on the object will shift and using a correlation procedure, the Kinect sensor can determine the distance of that object.

Various researchers have attempted to characterize the Kinect sensor and determine the optimal range to use it. Khoshelham and Elberink [4] explained the mathematical model of how the Kinect sensor determines the distance of an object and how to align the depth and RGB images. They concluded, using the OpenKinect framework and the RANSAC plane fitting method with the standard deviation of the residuals on the point cloud data, that the optimal distance to use the Kinect sensor is within 1000 - 3000 mm. At greater distances, the low resolution and noise of the sensor will reduce the accuracy of the data.

Alnowami et al. [12] observed a nonlinear relationship between the true depth and the Kinect pixel depth intensity, and they determined that the optimal performance lies between 800 mm and 1500 mm using the Microsoft SDK. Molnar et al. [27] reported a standard deviation of less than 1 mm to up to 10 mm at close ranges and between 7 and 50 mm at 3500 mm using OpenNI drivers. They found the optimal range of the Kinect sensor to be between 700 and 2000 mm. Andersen et al. [2] showed that the depth estimates of a pixel compared to the actual distance follow a linear relationship after being linearized by the OpenNI framework. The resolution of the Kinect was shown to be nonlinear and continues to degrade as the distance from an object increases.

Macknojia et al. [28] compared the performance of a K4W and a Kinect for the Xbox 360. They used the Microsoft SDK for the K4W sensor and the OpenNI framework for the Xbox Kinect sensor. They found that both devices have close to the same quantization error (the Xbox Kinect slightly overestimates it) and the optimal operating range was up to 2000 mm with an error of 10 mm. They also discovered that both Kinect sensors could not reconstruct transparent surfaces (since glass and clear materials are IR transparent), but could reconstruct curves and shiny painted surfaces on a car. Stommel et al. [29] describe a method in which missing depth values are estimated to eliminate gaps in images.

Stoyanov et al. [30] compared the Kinect against two ToF cameras (SwissRanger SR-4000 and Fotonic B70) to a laser range sensor (SICK LMS-200) and found that in short distances (< 3m), the Kinect sensor performed the closest to the laser range sensor. Smisek et al. [31] also compared the Kinect to two different sensors, a SwissRanger SR-4000 ToF sensor and a 3.5 M pixel SLR stereo camera. They concluded that the Kinect was much better than the ToF camera and comparable to the stereo camera.

Mallick et al. [32] review the work that has been done on the characterization of the noise in the Kinect sensors depth image. They conclude that the three main types of noise that occur are spatial, temporal, and interference noise. Spatial noise happens in a single frame, temporal noise happens over multiple frames, and interference noise happens when two or more Kinects are looking at the same scene.

The aim of this research is to compare the performance of different Kinect models using the OpenNI framework. OpenNI was chosen because the alignment of the depth and color images is performed automatically; it works with all the Kinect models, and is cross platform. In some of the tests reported in this paper, we also used the Microsoft Kinect SDK 1.8. The Kinect sensor will also be used to reconstruct 3-D objects of known dimensions.

The remainder of this paper is organized as follows. The next section discusses the methods used to characterize the different sensors. This is followed by Section III which shows and discusses the results obtained. Section IV discusses the 3-D reconstruction of gauge and machine blocks.

II. METHODS

A. Accuracy, Repeatability, and Resolution

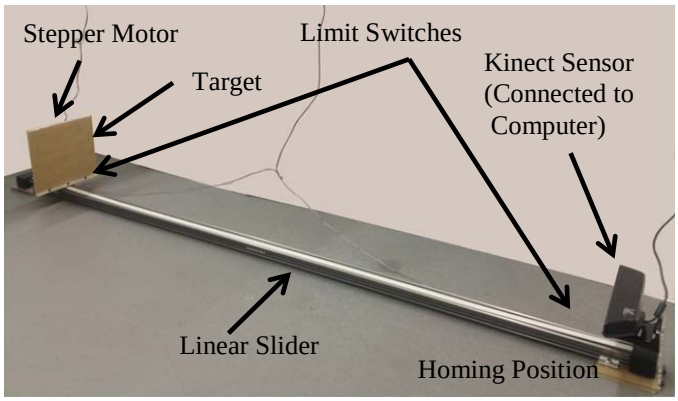

Fig. 1. Experimental setup of the linear slide used for the accuracy, repeatability, resolution, and ANOVA trials.

The experimental setup is shown in Fig 1. A 254 x 254 mm target machined out of plastic and covered with paper to make it non IR-transparent was attached to a stage mounted on a 2 m DryLin linear slide (Igus). A NEMA 23 stepper motor **(**1.8 deg/step, Minebea Co.**)** was attached to the linear slide. The stage was controlled by a data acquisition card (PCIM-DAS1602/16) which moved the target back in 100 mm increments. The Kinect sensor was mounted on one end of the

linear stage using a Kinect wall mount. This allowed for the model of the Kinect sensor to be interchanged while ensuring positional accuracy. The stepper motor and linear stage were calibrated by a displacement gauge (Mitutoyo) and a linear distance of 100 mm yielded an accuracy of +/-0.01 mm. When a test was started the stage would move towards the Kinect sensor to a homing position located 100 mm in front of the Kinect. The homing position was reached when the limit switch was triggered. Then the linear stage would move back in 100 mm intervals and the Kinect sensor recorded the depth of the target at every interval. The accuracy of each sensor was measured by constructing an area of 50x50 pixels in the center of the target. Twenty five images at 30 FPS were taken and the depth value of every pixel in the 50x50 pixel area was averaged to obtain a final value. For each interval, a total of 62500 pixels were used (25x50x50). The distance returned by the Kinect is the distance from the object to the plane of the Kinect's camera. An area of 50x50 pixels was chosen so the region of interest that the Kinect was looking at would be at the center of the target at the closest and farthest positions of travel. The repeatability of each sensor was measured by the standard deviation of these final values. A test was completed once the limit switch opposite the Kinect sensor was triggered. Distances greater than 1800 mm were not considered because of the physical limitations of the slider used. Also distances less than 600 mm were not explored because 500mm is very close to the limit of the Kinect in determining distances. The resolution step is the smallest increment that the Kinect can see at a distance. To determine the resolution of a Kinect sensor, pixels in the upper left, center, and lower right portion of the target were chosen and their values were recorded over 5000 images. The initial pixel value was subtracted from the subsequent pixel values and histograms of the differences were used to determine the resolution step. After a Kinect was placed on the holder, the distance from each end of the Kinect was measured to the target to ensure the Kinect was parallel to the target.

B. Steady State Temperature

To determine the amount of time it took a Kinect sensor to reach an internal steady state temperature, a temperature sensor (LM35) was connected to an Arduino Uno and interfaced to Matlab 2012a. The LM35 temperature sensor is a precision integrated-circuit that outputs an analog voltage proportional to the temperature in centigrade. This sensor does not require any external calibration and is accurate to ±1/4°C. The temperature sensor was attached directly above the infrared camera projector and LED on top of the plastic casing. This location was chosen because it was the location on the Kinect that heats up when the sensor is in use. The Kinect sensor was turned on and the OpenNI example program SimpleViewer was run for the duration of the test. The results of the trials were averaged, and the time constant (τ) was determined to be 1500 seconds. For the system to reach temperature stability, the Kinect sensor has to run for 4τ or ~6000 seconds (100 minutes). Fig. 2 shows the temperature response of all three Kinect models. This operating temperature will be referred to as the high temperature in the results, while the low temperature will be referred to as the room temperature the Kinect is at before being powered on which for this experiment is ~21-23°C. The Kinect sensor was set up to look at a stationary target 1800 mm away and images were recorded at 30 FPS over a period of 1200 minutes. A 50x50 pixel area was averaged over 25 images to obtain a depth distance. This test was repeated using the OpenNI driver and Microsoft Kinect SDK 1.8. The results of Fig. 2 will be discussed in Section III.

C. Statistical Methods

The design used for the experiment was a full factorial design whose effects model is shown in (1). Here, μ is the overall mean effect, τ_i is the effect of the *i*th level of the Kinect model and β_j is the effect of the *j*th level of the operating temperature. $(\tau\beta)_{ij}$ is the interaction effect between the Kinect model and the operating temperature, and (ε_{ijk}) is the random error term. y_{ijk} is the distance returned by the Kinect sensor when the Kinect model is at the *i*th level, the operating temperature is at the *j*th level and it is the *k*th replicate. Using Minitab's DOE tool, a random run order for the experiment was generated and a total of 18 tests were performed.

$$y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ijk} \qquad (1)$$

An ANOVA test (p= 0.05) was performed in Minitab on the response (depth data) of the Kinect sensor at 600 to1800 mm in 100 mm increments. This was used to examine if the Kinect model, the operating temperature, and/or and the interaction between the Kinect model and operating temperature were significant factors that influence the depth data. The ANOVA model was verified using Bartlett's test for equal variance (p=0.05) and checking the normality of residuals (p=0.05). A comparison of the means was performed using Tukey's test with a 95% confidence interval and the final recommendations were made by looking at the interactions plot.

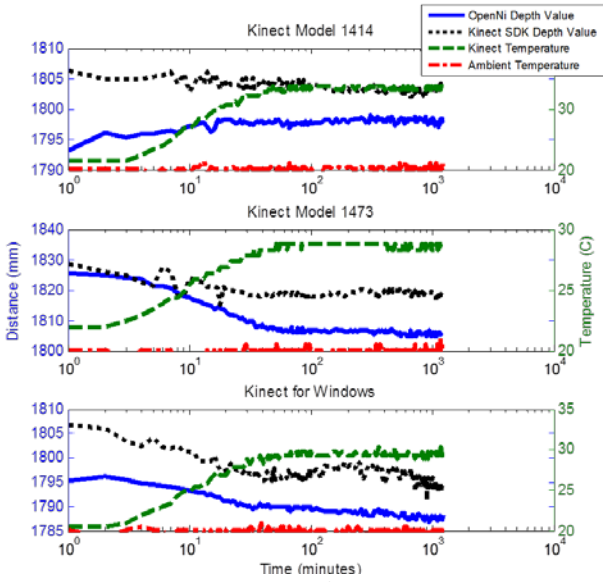

Fig. 2. Temperature response of the three Kinect models.

D. 3D Image Reconstruction

Reconstructing objects and locating holes is an important step in automated vision guided assembly and disassembly operations. In a disassembly operation, the vision system needs to be able to identify holes in order to try to remove a screw. It also requires precise reconstruction of objects and holes that might not always be flat or completely intact.

The Kinect model used for image reconstruction was the 1414 model which was mounted on a camera tripod on the top of a leveled surface 600 mm away from the object as shown in Fig. 3. This Kinect was chosen because it performed the best at close distances. Closer distances are better for applications involving precise reconstruction because the resolution of the Kinect is lower (1mm). These results are talked about more in Section III. Thus the purpose of the work in this section is to investigate how well the Kinect could reconstruct and identify features on these objects. The intent is not to perform comparison between different Kinect models.

The Kinect sensor was then leveled using its depth image, making sure all sides of the image returned the same depth value. Using the 150 x 90 pixel RGB image in Fig.4 the amount of pixels a 101.6 mm gauge block took up could be determined by subtracting the pixels of both sides of the gauge block on the same row (X_2 - X_1 where $Y_1 = Y_2$.) A millimeter to pixel (mm/px) resolution of 1.116 was calculated for this setup. Using the built in Matlab function "surf" and various sized gauge blocks, the Kinects sensor ability to accurately reconstruct 3D objects was tested. The experimental volumes of the gauge blocks were determined by separating the background from the gauge block using a height threshold.

Gauge blocks were chosen to be the objects reconstructed by the Kinect sensor because they are manufactured to known dimensions and tolerances. This makes them ideal to test the performance of the Kinect sensor.

Machinist blocks with a dimension of 25.4 x 50.8x 76.2 mm and containing six 12.7 mm through holes were used to test the Kinect sensors ability to reconstruct an angle an object was at and an object containing holes. The machinist blocks were angled using a height support and the angle was verified with a protractor. The angle of the machinist block was experimentally determined by using the depth image to obtain the y_1, y_2, z_1, and z_2 points shown in Fig.5. The angle of the machinist block could then be calculated using trigonometric relationships.

The Hough circle transform method (Tao Peng, Matlab Central File Exchange) was performed on the RGB image to obtain the location of the center of the holes for each of the six holes on the machinist block. The labels of these holes are given in Fig. 6a. Fig. 6b shows the calculation of the offset (r) between the between the center of the hole (A) and the center of the circle found with the circle transform algorithm (*A'*). This offset is given by (2). The offset was calculated with and without adjusting for the Kinect sensors position being directly above to the machinist block and a rotation in the X-Y plane. The theoretical locations of the center of the holes are determined by using the mm/px resolution, the dimensions of the machinist block, and the row of pixels in the RGB image

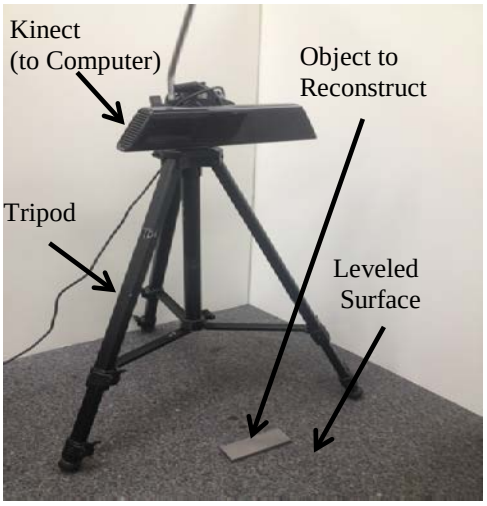

Fig. 3. Experimental Setup for 3D image reconstruction of gauge and machinist blocks.

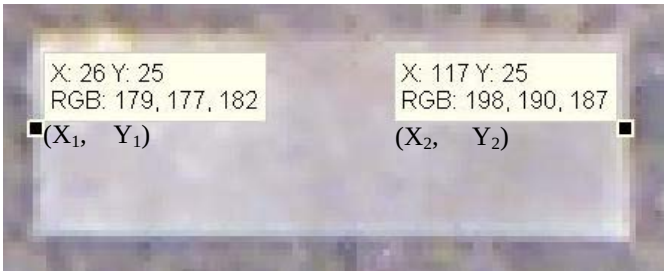

Fig. 4. Gauge block used for determining pixel to mm ratio. The X and Y values are used to determine the length of the gauge block in pixels.

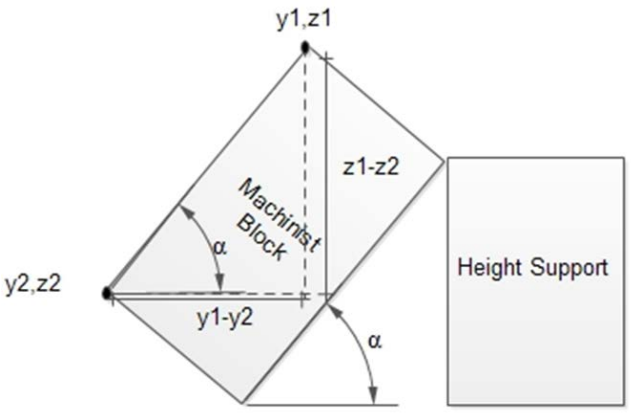

Fig. 5. Schematic showing how the angle of the machinist block is determined.

that the top of the block is at. These center hole locations work well for a flat object but as soon as the object is at an angle, the adjacent values A_1, A_2, and A_3 in Fig. 6c become a more accurate estimation of the location of the centers. Using similar triangle relations, the equations for the A_i values are given by (3) while the H_i values are found from the dimensions of the machinist block. The machinist block used in this setup has a hypotenuse of 7.62 mm.

The other adjustment, shown in Fig. 6d is performed to offset any rotation that the machinist block may have undergone in the x-y plane of the image. Since the locations of

the center of the holes are determined by their distance from the side of the block (in px), O_1, O_2, and O_3 allow for a more accurate starting pixel point. These points are also determined using similar triangles relations shown in (4). The values for H_i are found by using the dimensions of the machinist block. Fig. 6d only shows this adjustment on the side of the machinist block however, the same procedure will work for the rotation of the top part of the machinist block.

Using the depth image obtained from the Kinect sensor, at the location of the center holes, the amount of pixels in a 12x12 square that have a depth value of zero are added up. The size of the square was determined by calculating the pixel equivalent of the area of a 12.7 mm hole. This can be used to tell if there are patches of 0 depth pixels. These patches of pixels with 0 depth values can be used to validate a hole at these locations.

$$\Delta x = x_2 - x_1 \qquad (2)$$
$$\Delta y = y_2 - y_1$$
$$r = \sqrt{(\Delta x^2 + \ \Delta y^2)}$$

$$A_i = H_i \cos(\alpha) \quad where\ \alpha = \cos^{-1}(\frac{d_y}{H}) \qquad (3)$$

$$O_i = H_i \ \sin(\beta) \ \ where\ \beta = \sin^{-1}(\frac{d_x}{H}) \qquad (4)$$

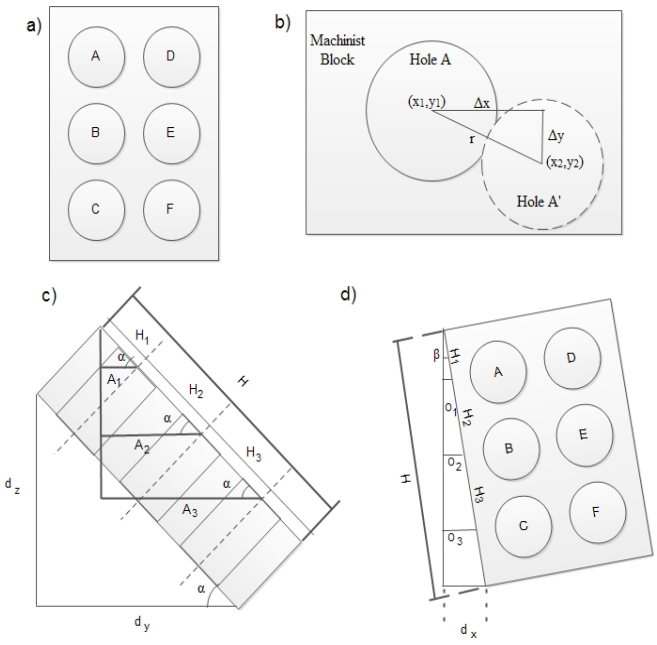

Fig. 6. a) Labeling of the holes on the Machinist Block b) Calculation of the offset between the hole and circle from the image c) Correction for the way the Kinect is positioned over the machinist block d) Correction for rotation of machinist block.

III. Results

A. Accuracy and Repeatability

Each model of the Kinect sensor was used for three trials at a high and a low temperature as part of the ANOVA test (Section C). The averaged results of all three Kinect models are shown in Fig. 7. This figure plots the actual distance the target is from the Kinect sensor vs. the distance calculated by the Kinect sensor at different temperatures. The standard deviations of all the points from these trials with all the Kinect models are very small with the greatest standard deviation being just over 2mm (Kinect 1414 at 1700mm). Due to the small size of the standard deviation and the scale of the graph these error bars are not shown. At low temperatures (~21 ºC - temperature when Kinect was just turned on) and close distances less than 800 mm, all the calculated results are close to the actual distances (within 2.1mm), but at greater distances than 1300 mm, the Kinect model 1473 starts to calculate distances farther away than the actual distance. At a high temperature (temperature after turning on the Kinect for more than 100 minutes), the Kinect 1473 improves the distances it reads as the target gets farther away while the other two Kinect models get worse and underestimate the depth.

B. Depth Resolution

The depth resolution histogram of each Kinect sensor was obtained at different distances. A histogram of the resolution at 600mm with the Kinect model 1414 is shown in Fig. 8. This figure shows a resolution of 1 pixel. These histograms are summarized in Fig. 9. As the distance from a target increases, the resolution of the Kinect sensor for all models becomes coarser. The most accurate resolutions occur at the closer distances (distances < 700 mm) when the resolution of all the models are 1 mm. At 1800 mm, the resolutions of the models are still less than 10 mm. All of the different Kinect models exhibit the same upward sloping graph and the same resolution. The results presented agree with the results presented by Andersen et al. [2] and Macknojia et al. [28] at these distances. Andersen et al. [2] reported approximately a 1 mm resolution at a distance of 600mm to around a 10 mm resolution at a distance 1800mm. Macknojia et al. [28] also reported a 1 mm resolution at a distance of 600 mm to 9 mm at a distance of 1800 mm. Both of these authors also demonstrated that at distances of around 3000mm, the resolution can be as poor as 30 mm. This was not explored with this setup due to the size of the linear slider.

C. ANOVA Testing

For all of the distances, the *Kinect Model* and the *Temperature* were significant factors with *p*-values of 0.000. The *p*-value of the interaction between these two factors was a significant (*p* value <0.05) at all distances except for 600mm where the *p*-value was 0.089. However, since this value is close to 0.05 and can be interpreted as moderately significant, it was determined to treat the interaction of this distance as significant.

The results for the comparison of the means test for the Kinect models are shown in Table I. Since there was interaction between the *Kinect Model* and *Operating Temperature* was taken to be a significant factor at all the distances, the comparison of the means had to be performed

for both the low and high temperatures.

In Table I, the grouping category shows how the response of the different Kinect models compared with each other. If the response of the Kinect models were not statistically different, then the letter under grouping would be the same. If the responses of the Kinect models were statistically different, then the letters would be different. For example, at 600 mm under the *Temperature Low* column, the response of the Kinect model 1473 and K4W are not statistically different from each other, thus both have a grouping of "A". The Kinect model 1414 is statistically different from both of the other models so it has a grouping of "B."

Table I shows that in 11 of the 26 trials all of the Kinect sensors have a response that is significantly different from each other. It also shows that in 7 of the 26 trials the response of the Kinect 1414 and K4W are not significantly different while the response between them and the Kinect 1473 is. The Kinect 1414 and K4W are more likely to produce results that are not statistically different at distances when the target was farther away (<1200mm) at a low temperature and (<1700mm) at high temperatures.

Table II provides a summary of the Interaction plots as well as showing the most consistent performing Kinect model. In the second row labeled *Kinect Model (ANOVA)*, the table shows the model that performed the "best" when the target was at different distances from the Kinect.

The row labeled *Temp (ANOVA)* shows at what temperature the Kinect performed the "best" at. The row *Best Model (ANOVA)* is used to show that the Kinect 1414 performs the best at close distances (<900mm). It also shows the Kinect

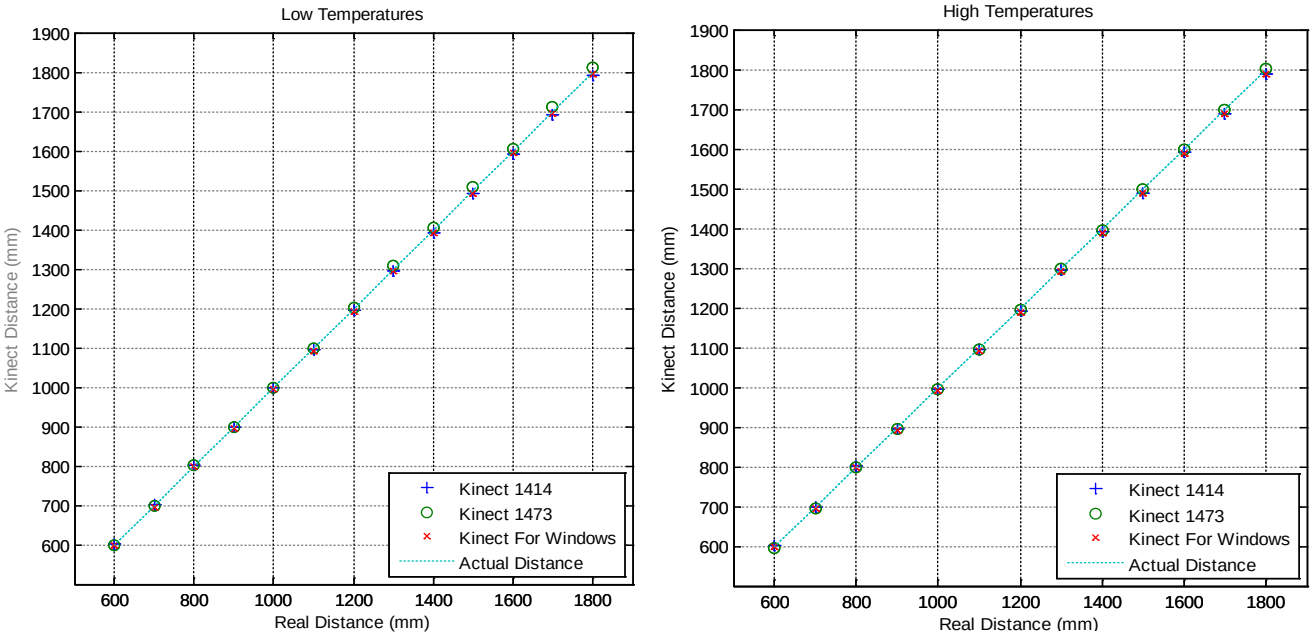

Fig. 7. Accuracy graph of the Kinect sensor at high and low temperatures (Actual distance of object vs. distance returned by Kinect sensor)

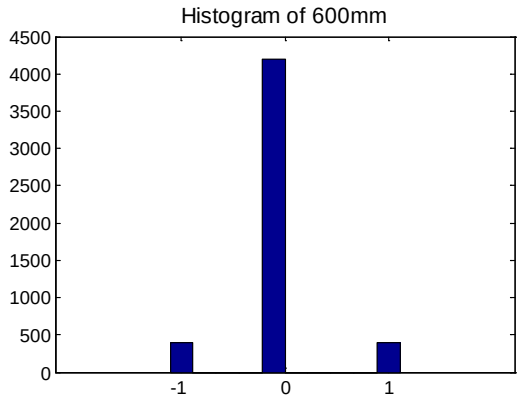

Fig. 8. Histogram of the resolution data from 600mm.

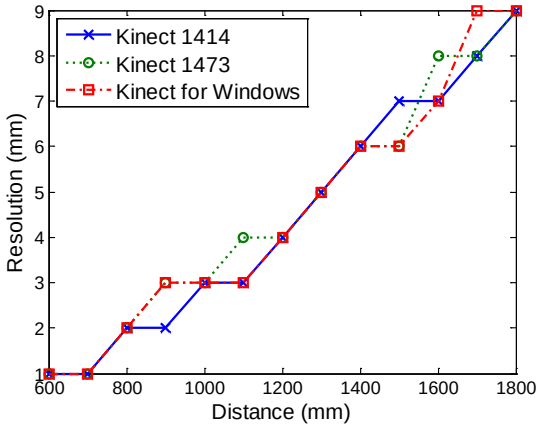

Fig. 9. Resolution of different Kinect sensors.

1473 performs best at farther distances and performs the most consistently over the most distances. Both the Kinect 1414 and 1473 have distances where they performed best at low temperature and distances where they performed best at high

temperatures. There are two models listed under 900mm because according to Table II, there is no significant difference between these two models at a low temperature.

Fig. 2, shows that as the temperature of the Kinect increases, the depth response continues to change. It also shows that the response stabilizes once the Kinect reaches a high steady state temperature. For all three Kinect models, the Microsoft Kinect SDK returned a depth distance that was farther away than the depth distance returned by OpenNI. The Kinect model 1414 had its depth response increase when using the OpenNI drivers and decrease while using the Microsoft Kinect SDK. For the Kinect models 1473 and K4W, using either driver the depth response decreased and then reached a steady state as the Kinect's temperature increased and reached steady state. Using the OpenNI drivers, the depth response of the Kinect 1473 changes about 20mm while the depth response of the Kinect 1414 differs about 5mm as the Kinect goes from a low to a high temperature. The K4Ws depth response changes about 6 mm from a low temperature to a high temperature but after about 350 minutes the depth response begins to drop again. Using the Microsoft Kinect SDK the magnitude of the depth response change is about 10mm Kinect 1473 and is about 3mm for the Kinect 1414. The K4W depth response changes about 10 mm and begins to drop again after about 250 minutes.

Since the response of the Kinect is more stable at a high steady state temperature, the row *Best Model at High Temp* in Table II shows which Kinect Model performs the best at a high temperature. Again, the Kinect model 1414 performs the best at close distances while the Kinect 1473 performs the best at farther distances.

All of the results for the distances shown in Table I and Table II satisfied the normality and equal variance assumption with a *p*-value above 0.05.

IV. 3D Reconstruction

A. Gauge Blocks

Fig. 10a shows a single frame of the gauge block as captured by the Kinect sensor which was mounted 600mm away. The figure shows that there is a distortion in the image. The left and right spikes shown in the image are pixels whose depth values are zero. Applying a correction factor improved the image (See Fig. 10b). Looking at the pixel depth values on the y-axis in Fig 10a, a *Height Difference* of around 8 mm is determined. On the left side of the image, the depth value is around 391 mm and on the right side around 399 mm (It should be noted this value has been subtracted from 1000 in order to invert the image for viewing. The distance from the Kinect to the table was 600 mm). A linear function was developed to be used as a correction factor to flatten out the image that allows for a better estimate of the volume of the gauge block. This method works the best when it is possible to distinguish the background from the object(s) of interest. The correction formula is given by (5). In this equation, *i* represents the row of the image, and *j* represents the column of the image. The correction factor (*CF*) from (5) given by (6). The negative sign (-) in (6) shifts the depth values up to the correct height. In (6), N is the last column. In Fig. 10, 150 columns of the image were adjusted therefore, N would be 150.

TABLE I
TUKEY RESULTS AND FOR COMPARISON OF THE MEAN OF DIFFERENT KINECT MODELS

Temperature Low							
Distance (mm)	600	700	800	900	1000	1100	1200
Model							
1473	A	A	A	A	A	A	A
1414	B	B	A	A	B	B	B
K4W	A	C	B	B	C	C	B
Distance (mm)	1300	1400	1500	1600	1700	1800	
Kinect							
1473	A	A	A	A	A	A	
1414	B	B	B	B	B	B	
K4W	B	B	B	C	B	C	
Temperature High							
Distance (mm)	600	700	800	900	1000	1100	1200
Model							
1473	A	A	A	A	A	A	A
1414	B	B	B	B	A	A	B
K4W	A	A	A	C	B	B	C
Distance (mm)	1300	1400	1500	1600	1700	1800	
Model							
1473	A	A	A	A	A	A	
1414	B	B	B	B	B	B	
K4W	C	C	C	C	B	B	

TABLE II
SUMMARY OF INTERACTIONS PLOTS

Distance (mm)	600	700	800	900	1000	1100	1200
Kinect Model (ANOVA)	1414	1414	1414	1473	1473	1473	1473
Temp (ANOVA)	High	Low	High	Low	Low	Low	Low
Best Model (ANOVA)	1414	1414	1414	1414/ 1473	1473	1473	1473
Best Model High Temp	1414	1414	1414	1414	1473/ 1414	1473/ 1414	1473
Distance (mm)	1300	1400	1500	1600	1700	1800	
Kinect Model (ANOVA)	1473	1473	1473	1473	1473	1473	
Temp (ANOVA)	High	Low	High	High	High	High	
Best Model (ANOVA)	1473	1473	1473	1473	1473	1473	
Best Model High Temp	1473	1473	1473	1473	1473	1473	

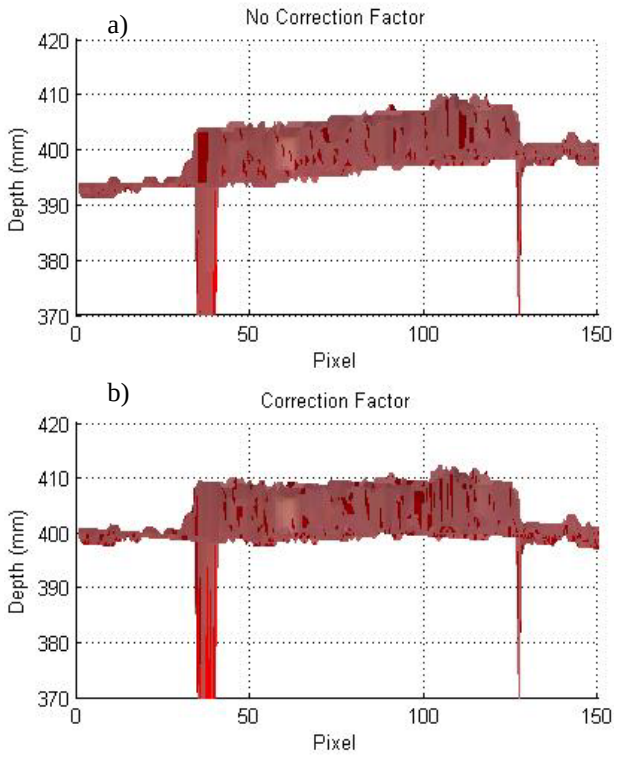

Fig. 10a. Image of gauge block without correction factor. b. Image of gauge block with correction factor applied. This image was taken with the Kinect set up 600mm away from the gauge block.

$$Image(i,j) = Image(i,j) + CF \tag{5}$$

$$CF = \left(\frac{Height\ Difference}{N} * j - Height\ Difference\right) \tag{6}$$

Fig 11. shows the effect different correction factors and cutoffs have on the percent error in estimating the volume of different sized gauge blocks. Two parameters are shown in the legend in the figure. The first value is the *Height Difference* (in pixels) from (5) and the second value is the *Cutoff Height* (in mm) used to determine where the background ends and the object begins for the setup used in this experiment. Trying to determine the volume of an object without a correction factor led to an underestimation of the objects volume. This is due to the fact part of the object would be beneath the cutoff height and the program would perceive it as background. By introducing a correction factor, the ability to accurately reconstruct an objects volume greatly increases. The most consistent combination of *Height Difference* and *Cutoff Height* is 6 and 598 mm for this geometry. By using these values, all the percentage errors are lower than 10 percent. The percent error also decreased as the length of the gauge blocks increased. When the 101.6 mm gauge block was analyzed, the percent error of the volume was under 10 % regardless of the *Cutoff Height* and *Height Difference* values. Fig. 11 shows there is a genuine need to calibrate the Kinect's output in order to get meaningful data while trying to reconstruct a 3-D object. These correction factors show that the output from the Kinect sensor needs to be calibrated for fine 3D reconstruction and cannot just be used as is.

B. *Machinist Blocks*

1) *Angle*

The angle the machine blocks were setup at versus the angle calculated by the Kinect sensor are shown in Fig. 12. The angles are plotted with and without using a correction factor. All of the angles calculated just by using the depth image, are less than the actual angle that the machinist block was setup at. An average correction factor for the angle in this setup was determined to be 1.3708 (R^2 value of .9957). This correction factor was determined by dividing the actual angle by the depth angle. Multiplying the correction factor by the previous angle greatly reduces the percent error between the actual angle and the angle calculated. In Fig. 12, all of the adjusted angles are more accurate than the unadjusted angles. These results agree with the results presented by Tahavori et al. [14] in which they showed the Kinect underestimated the angles.

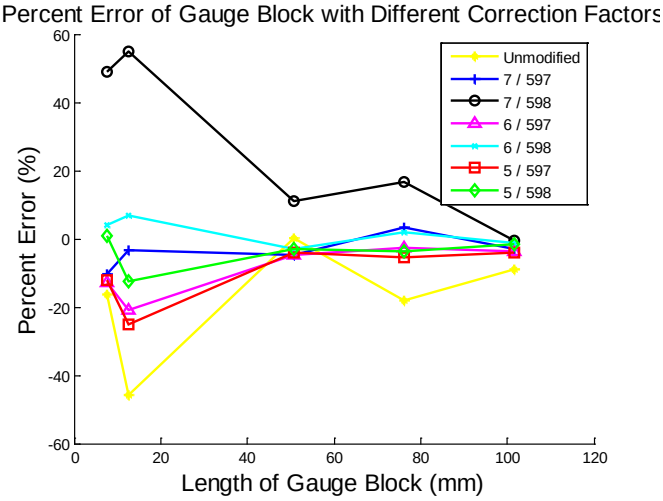

Fig. 11. Percent error of volume vs. length of gauge block in gauge block reconstruction.

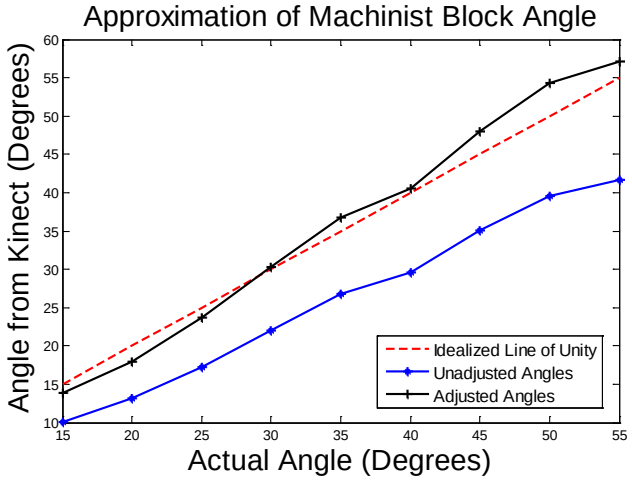

Fig. 12. The unadjusted and adjusted machinist block angles determined by the Kinect sensor

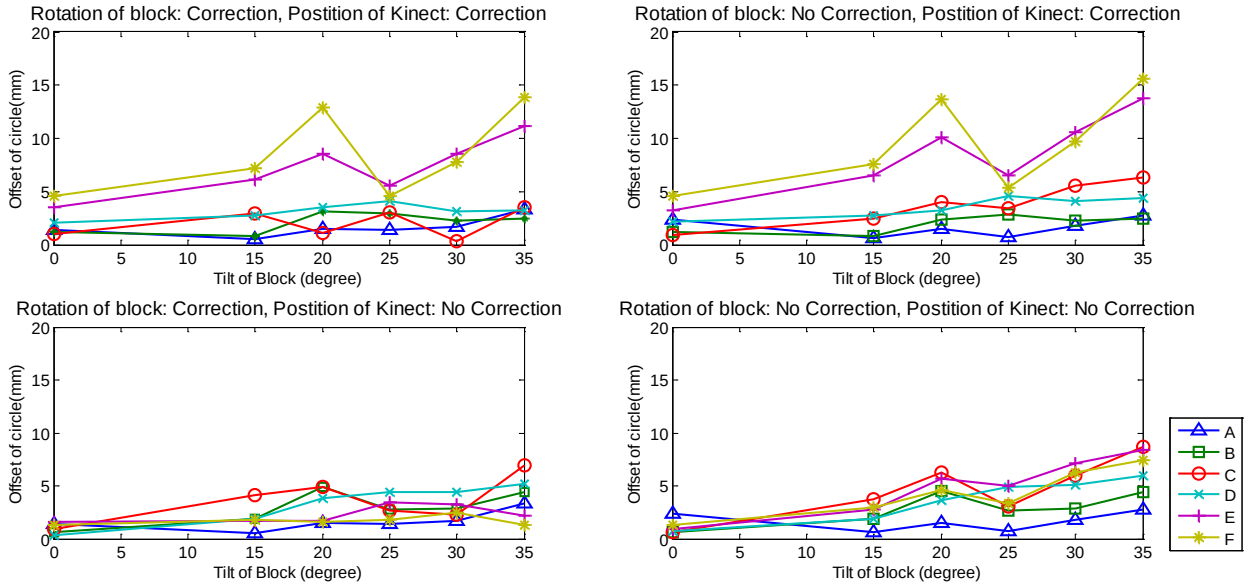

Fig. 13. Percent error of volume vs. length of gauge block in gauge block reconstruction.

2) Holes

The offset of the center of the circles found with the circle detection algorithm from the centers of the hole in the machinist blocks are shown in Fig 13. These results show that correcting for the position of the Kinect sensor using (3) does not improve the accuracy of this offset value. However, correcting for a rotation of the machinist block using (4) does improve the offset. The best combination (lowest offset) is found correcting for the rotation of the machinist block but not correcting for the position of the Kinect sensor. With these parameters, the offset between the centers of the circles for all of the holes are less than 10 mm. As the angle of the machinist blocked increases, the offset between the two circles also increases. This method worked for the machinist blocks when the angle of tilt was less than 35 degrees. When the machinist block was positioned at an angle greater than 35 degrees, the circle detection algorithm was unable to determine the circles for the holes as they become elliptical.

The amounts of pixels returning a depth value of 0 at the center of the circles are shown in Fig 14. This figure shows that clusters of pixels with a depth of zero are around all of the holes of the machinist blocks (labeled A-F). At lower angles, there is a higher pixel count around the holes than at higher angles which signifies that the Kinect is able to determine there is a hole in that location. However, this method only worked when the angle of the machinist block was less than 40 degrees. When the machinist blocks were placed at angles greater than 40 degrees, the depth images from the Kinect sensor were not reliable, as parts of the depth image of the machinist block were missing and thus had pixels with depths values of 0.

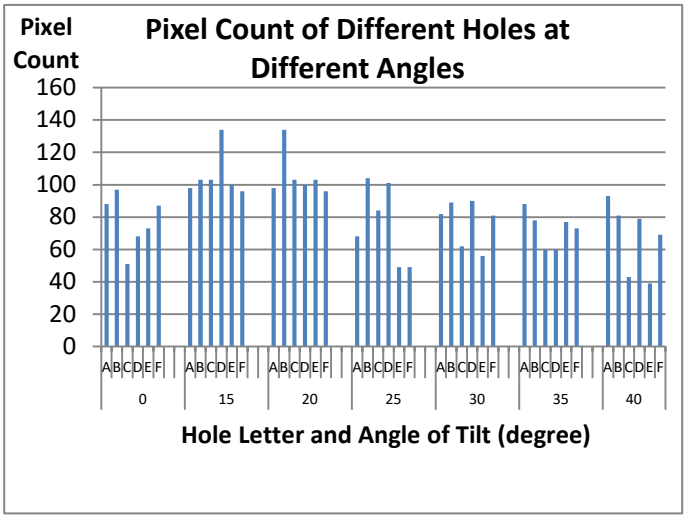

Fig. 14. Zero depth pixel clusters around the location of holes in the machinist blocks. The letters A-F in the legend, correspond to hole location.

V. Conclusions

Overall, the Microsoft Kinect sensor is an inexpensive sensor that is capable of producing acceptable results when determining the distance of an object from the sensor. All of the Kinect models give very accurate results at close distances (600 mm to 800 mm). At a low operating temperature, the Kinect 1473 overestimates the depth to an object as the distance increases. At a high operating temperature, the Kinect 1473 performs better as the target moves farther away (towards 1800mm) while the other two Kinect models perform worse and underestimate the distance. The standard deviation of all the Kinect sensors is low (< 2.1mm) at all the distances tested which shows the Kinect sensor has good repeatability.

The resolution of all the Kinect sensors are best at close

ranges (<700mm) as the resolution at these distances is 1 mm. At a distance of 1800 mm from the sensor, the resolution is still less than 10 mm. However this may make it unattractive for applications requiring precise measurements.

From the ANOVA study, the Kinect model and the operating temperature the Kinect sensor is at, and their interactions are both significant factors in the depth response. The interaction plots show that the Kinect model 1414 is the most consistent model at close distances (<1100mm) while the Kinect 1473 produces the best results at farther distances (>1000mm). The overlap in distances of 1000 and 1100mm between these two Kinects are because the depth response between the two models is not statistically significant.

It is also shown that the Kinect depth response changes with the temperature of the Kinect sensor and the software driver used. This response becomes more stable when the Kinect reaches a high steady state temperature. For all the Kinect models, the Microsoft SDK returned a depth distance at high steady state temperature that was further away than the depth returned by OpenNI. Therefore it may be beneficial to have the Kinect warm up before using it as this depth response change can be rather large.

The Kinect sensor is also able to accurately reconstruct a 3D object and determine the objects volume. However, it was important to use a correction factor and correctly pick a cutoff value to separate the background from the object. It is not a robust solution as a change of just one pixel in the cutoff value can drastically alter the results and is geometry dependent. It does however show that there is a need for calibration of the Kinect's output for fine measurements.

Using the Kinect depth image, the Kinect tends to underestimate the actual angle of an object. It is also possible to account for the angle that an object was setup at after applying another correction factor. The Kinect sensor was able to estimate the offset from the center of a circle found with a circle detection algorithm to the center of a hole in the machinist block. The best results were obtained by adjusting for a rotation of the machinist block while not adjusting for the position of the Kinect sensor above the machinist block.

Additionally clusters of pixels containing a depth value of zero can also be used to indicate a hole at certain angles provided the angles are not too steep (< 40 degrees). This would be important in automated computer vision applications. Both the RGB and depth images should be used in conjunction with each other in order to confirm the results and determine if a hole is actually present.

REFERENCES

[1] J. Han, L. Shao, D. Xu, and J.Shotton, "Enhanced Computer Vision with Microsoft Kinect Sensor : A Review," *IEEE Trans. Cybern*, vol. 43, no. 5, pp. 1318–1334, Oct 2013.

[2] M. R. Andersen, T. Jensen, P.Lisouski, A.K. Mortensen, M. K. Hansen, T.Gregersen, and P. Ahdrendt., "Kinect Depth Sensor Evaluation for Computer Vision Applications", Dept. Eng. Electr. Comput. Eng., Aarhus Univ., Aarhus, Denmark. Tech. Rep. ECE-TR-6, Feb. 2012.

[3] P. Henry, M. Krainin, E. Herbst, X .En,, and D. Fox "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments," *Int. J. Rob. Res.*, vol. 31, no. 5, pp. 647–663, Feb. 2012.

[4] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of Kinect depth data for indoor mapping applications.," *Sensors*, vol. 12, no. 2, pp. 1437–1454, Jan. 2012.

[5] N. Rafibakhsh, J. Gong, M. K. Siddiqui, C. Gordon, and H. F. Lee "Analysis of XBOX Kinect Sensor Data for Use on Construction Sites : Depth Accuracy and Sensor Interference Assessment," in *Construction Research Congr.*, May 2012, pp. 848–857.

[6] M. Tölgyessy and P. Hubinský, "The Kinect Sensor in Robotics Education," in *Proc.2nd Int. Conf. Robotics Educ.*, Sep. 2011, pp. 143–146.

[7] J. Stowers, M. Hayes, and A. Bainbridge-Smith "Altitude control of a quadrotor helicopter using depth map from Microsoft Kinect sensor," in *2011 IEEE Int. Conf. on Mechatronics*, Apr. 2011, pp. 358–362.

[8] R. A. El-iaithy, J. Huang, and M. Yeh, "Study on the Use of Microsoft Kinect for Robotics Applications," in *Position Location and Navigation Symp.*, Apr. 2012, pp. 1280–1288.

[9] S. Shirwalkar, A. Singh, K. Sharma, and N. Singh, "Telemanipulation of an industrial robotic arm using gesture recognition with Kinect," *2013 Int. Conf. Control. Autom. Robot. Embed. Syst.*, Dec. 2013. pp. 1–6,

[10] R. Afthoni, A. Rizal, and E. Susanto, "Proportional Derivative Control Based Robot Arm System Using Microsoft Kinect," in *2013 Int. Conf. Robotics, Biomimetics, Intelligent Computational Syst.*, Nov. 2013, pp. 25–27.

[11] L. Gallo, A. P. Placitelli, and M. Ciampi, "Controller-free exploration of medical image data : experiencing the Kinect," in *Int. Symp. Comput. Based Medical Syst.*, Jun. 2011, pp. 1–6.

[12] M. Alnowami, B. Alnwaimi, F. Tahavori, M. Copland, and K. Wells, "A quantitative assessment of using the Kinect for Xbox360 for respiratory surface motion tracking," in *Proc. SPIE Medical Imaging 2012: Image-Guided Procedures, Robotic Interventions, and Modeling*, Feb. 2012, vol. 8316. pp T1-T8.

[13] F. Tahavori, M. Alnowami, and K. Wells, "Marker-less respiratory motion modeling using the Microsoft Kinect for Windows," in *Proc. SPIE, Medical Imaging 2014: Image-Guided Procedures, Robotic Interventions, and Modeling* , Mar. 2014, vol. 9036, pp K1-K10.

[14] F. Tahavori, M. Alnowami, J. Jones, P. Elangovan, E. Donovan, and K. Wells, "Assessment of Microsoft Kinect Technology (Kinect for Xbox and Kinect for Windows) for Patient Monitoring during External Beam Radiotherapy," in *Nucl. Sci. Symp. and Medical Imaging Conf.*, 2013, pp. 1-5.

[15] R. A. Clark, Y. H. Pua, K. Fortin, C. Ritchie, K. E. Webster, L. Denehy, A. L. Bryant, "Validity of the Microsoft Kinect for assessment of postural control.," *Gait Posture*, vol. 36, no. 3, pp. 372–377, Jul. 2012.

[16] B. Lange, C.-Y. Chang, E. Suma, B. Newman, A. S. Rizzo, and M. Bolas, "Development and evaluation of low cost game-based balance rehabilitation tool using the Microsoft Kinect sensor," in *Proc 2011 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2011, pp. 1831–1834.

[17] X. Ning and G. Guo, "Assessing Spinal Loading Using the Kinect Depth Sensor: A Feasibility Study," *IEEE Sens. J.*, vol. 13, no. 4, pp. 1139–1140, Apr. 2013.

[18] Y. Yang, F. Pu, Y. Li, S. Li, Y. Fan, and D. Li, "Reliability and Validity of Kinect RGB-D Sensor for Assessing Standing Balance," *IEEE Sens. J.*, vol. 14, no. 5, pp. 1633–1638, May 2014.

[19] D. S. Alexiadis, P. Kelly, P. Daras, N. E. O'Connor, T. Boubekeur, and M. Ben Moussa, "Evaluating a dancer's performance using kinect-based skeleton tracking," in *Proc.19th ACM Int. Conf on Multimedia*, 2011, pp. 659–662.

[20] B. John and G. Fang, "Human Object Recognition Using Colour and Depth Information from an RGB-D Kinect Sensor," *Int. J. Adv. Robot. Syst.*, vol. 10, Jan. 2013.

[21] S. Obdrzálek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel, "Accuracy and robustness of Kinect pose estimation in the context of coaching of elderly population.," *in Proc. 2012 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2012, pp. 1188–1193.

[22] "History - OpenKinect." [Online]. Available: http://openkinect.org/wiki/History. [Accessed: 25-Sep-2013].

[23] "Microsoft Releases Kinect for Windows SDK Beta for Academics and Enthusiasts." [Online]. Available: http://www.microsoft.com/en-us/news/press/2011/jun11/06-16mskinectsdkpr.aspx. [Accessed: 25-Sep-2013].

[24] A. Kadambi, A. Bhandari, and R. Raskar, "3D Depth Cameras in Vision: Benefits and Limitations of the Hardware With an Emphasis on the First- and Second-Generation Kinect Models," in *Computer Vision and Machine Learning with RGB-D Sensors*, L. Shao, J. Han, P. Kohli, and Z. Zhang, Eds. Cham: Springer International Publishing, 2014, pp. 3–26.
[25] J. Sell and P. O. Connor, "The Xbox One System on a Chip and Kinect Sensor," *IEEE Micro*, vol. 34, no. 2, pp. 44–53, 2014.
[26] D. Um, D. Ryu, and M. Kal, "Multiple Intensity Differentiation for 3-D Surface Reconstruction With Mono-Vision Infrared," *IEEE Sens. J.*, vol. 11, no. 12, pp. 3352–3358, 2011.
[27] B. Molnar, C. K. Toth, and A. Detrekoi, "Accuracy Test of Microsoft Kinect for Human Morphologic Measurements," in *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, Aug. 2012, vol. XXXIX-B3, pp. 543–547.
[28] R. Macknojia, A. Chávez-Aragón, P. Payeur, and R. Laganière, "Experimental Characterization of Two Generations of Kinect ' s Depth Sensors," in *IEEE Int. Symp. Robotic and Sensors Environments*, Nov. 2012, pp. 150–155.
[29] M. Stommel, M. Beetz, and W. Xu, "Inpainting of Missing Values in the Kinect Sensor's Depth Maps Based on Background Estimates," *IEEE Sens. J.*, vol. 14, no. 4, pp. 1107–1116, Apr. 2014.
[30] T. Stoyanov, A. Louloudi, H. Andreasson, and A. J. Lilienthal, "Comparative Evaluation of Range Sensor Accuracy in Indoor Environments," in *Proc. 2011 IEEE Int. Conf Comput. Vision Workshops*, Nov. 2011, pp. 1154–1160.
[31] J. Smisek, M. Jancosek, and T. Pajdla, "3D with Kinect," in *Consumer Depth Cameras for Computer Vision—Research Topics and Applications: Advances in Computer Vision and Pattern Recognition.* New York, NY, USA: Springer-Verlag, 2013, ch. 1, pp. 3-25.
[32] T. Mallick, P. P. Das, and A. K. Majumdar, "Characterizations of Noise in Kinect Depth Images: A Review," *IEEE Sens. J.*, vol. 14, no. 6, pp. 1731–1740, Jun. 2014.

Nicholas M. DiFilippo received his MS degree in mechanical engineering from the University of Rhode Island, USA in Summer 2012. He is currently pursuing his Ph.D. degree in mechanical engineering at the same institute. His research interests include vision systems, robot motion planning, and motion control

Musa K. Jouaneh (M, SM) is Professor of Mechanical Engineering and the director of the Mechatronics laboratory at the University of Rhode Island (URI). He received his B.S. degree in Mechanical Engineering in 1984 from the University of Louisiana at Lafayette, and his Master and Doctorate degrees in Mechanical Engineering from the University of California at Berkeley in 1986 and 1989 respectively. His research interests include mechatronics and robotics with particular interest in motion control systems. He has served as a consultant to many companies in the Northeast and has received two College of Engineering Faculty Excellence Awards and the URI Foundation Teaching Excellence Award. He is the author or co-author of over 65 publications including two US patents. He is also the author of two recently published textbooks on Mechatronics. Dr. Jouaneh is a Fellow member of ASME, and a member of ASEE.