A DEPTH IMAGE QUALITY BENCHMARK OF THREE POPULAR LOW-COST DEPTH CAMERAS

VYOMKESH KUMAR JHA, STEFAN GRUSHKO, JAKUB MLOTEK, TOMAS KOT, VACLAV KRYS, PETR OSCADAL, ZDENKO BOBOVSKY

Department of Robotics, Faculty of Mechanical Engineering, VSB-TU Ostrava, Ostrava, Czech Republic

DOI: 10.17973/MMSJ.2020_12_2020057

e-mail: vyomkesh.kumar.jha@vsb.cz

A depth camera outputs an image in which each pixel depicts the distance between the camera plane and the corresponding point on the image plane. Low-cost depth cameras are becoming commonplace and given their applications in the field of machine vision, one must carefully select the right device according to the environment in which the camera will be used given the accuracy these cameras can be associated with factors such as distance from the target, luminosity of the environment, etc. This paper aims to compare three depth cameras currently available in the market, Intel RealSense D435, which uses stereo vision to compute depth at pixels, ASUS Xtion and Microsoft Kinect 2 represent Time of flight-based depth cameras. The comparison will be based on how the cameras perform at different distances from a flat surface and we will check if the colour of the surface affects the depth image quality.

KEYWORDS

Sensor Comparison, Depth Images, Image Processing, Image Quality

1 INTRODUCTION

Depth cameras are becoming widely available from multiple vendors such as Intel, Asus, Microsoft, Photoneo, etc. Such wide availability at reasonable prices is opening up scope for new applications in different use cases as discussed in [Francis 2015], [He 2018] and [Nock 2013], which makes it necessary to make more information about the cameras' performance in real-world applications [Chuang-Yuan Chiu 2019] than the manufacturers publish available to the general public. Several techniques have been employed in the past to measure the quality of the depth images produced by these cameras. In [Langmann 2012] paper, techniques pertaining to the usage of Bohler star [Böhler 2003] which is a tool used to determine the angular or lateral resolution of depth measurement devices, he also uses surfaces with different shapes to measure the depth resolutions of cameras at different distances. His work is the closest to this work. Another work done in this field is [Lottner 2008] in which they discuss the impact of lightning on the resulting frames. We derived some information from [Swoboda 2014] article on characterisation of the Asus Xtion Pro depth sensor. The real benefit of these measurements is their use in predicting depth frame error models for the cameras [Bobovský 2018], one such work is [Sweeney 2019].

2 DESIRED OUTCOME

In this article, Microsoft Kinect 2, Asus Xtion 2 and Intel RealSense D435 [Tadic 2019] cameras are compared, the specifications of interest are mentioned in Table 1.

Our experiments are designed to measure the error in each frame while varying the distance between the imaged surface and the camera in an indoor environment [Kazmi 2014], keeping the camera plane parallel to the imaged surface. The colour of the imaged surface was varied to check for its effects on the resulting frame. We used coloured paper with a non-reflective texture to test for the camera's performance while imaging different colours. The experiment process involved interfacing the cameras to the computer running the data collection program for which we developed a C++ library called libUniCam which encapsulates different libraries needed to access the individual cameras and allows the developers to utilise a common interface to access any camera. We also implemented a special mechanical structure to accurately position the camera at any distance from the plane it images and a stabiliser for the camera to ensure that the image plane (Figure 1) is parallel with the imaged plane.

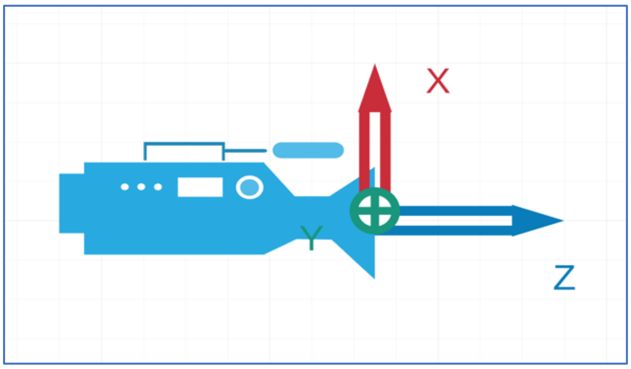

Figure 1. XY plane is the image plane.

Table 1. Information about the tested Cameras

Camera	RealSense D435	ASUS Xtion 2	Kinect 2
Resolution	640*480	640*480	512*424
Imaging Range [m]	10	0.8-3.5	0.5-4.5
Expected Error [%]	<=2%	<=2%	0.56%
Library Used	LibRealSense 2	Asus OpenNI 2	Libfreenect 2
Type of Sensor	Structured Light	Time of Flight [Rapp 2008]	Time of Flight

3 TESTING APPARATUS

To ensure that the measurements stayed consistent, we made a rigid frame fixed to one axis of motion towards and away from the wall (Z axis of camera) (Figure 3). To get the best results, a hardware stabiliser made using 2 servo motors, controlled with the help of an Arduino board communicating with the computer was implemented (Figure 2). The stabiliser communicates with the PC which processes the depth frame coming from the camera to computer if the wall is normal to the camera or not and sends instructions to the Arduino to reorient the camera if needed. Since we had to communicate with multiple cameras for the experiment, we wrote a wrapper around the camera libraries that allowed us to access any of cameras used, using a unique interface allowing us to use one software to for all the measurements.

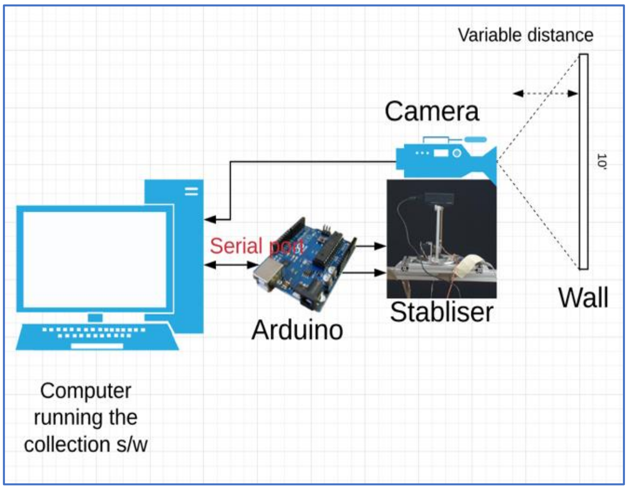

Figure 2. Block diagram of the experimental setup.

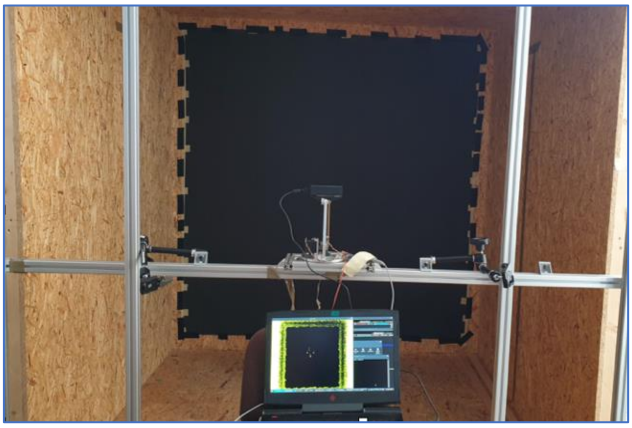

Figure 3. The testing apparatus.

4 EXPERIMENTS

To get meaningful results about the accuracy of the depth frames, the cameras were placed facing towards the wall, 50 frames were collected for every distance between 600 mm to 2300mm every 100 mm and we performed the measurement process 10 times to reduce experiment induced errors, effectively yielding us with 500 frames for every distance from the wall, for every camera, and for 5 chosen colours (Red, Blue, Green, Black, White).

4.1 *Aligning the camera to the wall*

To ensure the camera's Z axis is perpendicular to the wall, we take a depth frame and then select two rectangular regions (Figure 4) equidistant from the centre axis (for both vertical and horizontal axes), average the distance of the depth pixels inside the region and calculate the difference (disparity) between the averages (Figure 5), this difference should be close to 0 if the camera is aligned the correct way as both the rectangular regions would be equidistant from the camera. Depending on the axis corresponding to which this disparity was calculated and if it's positive or negative, we rotate the camera around the corresponding axis in the direction that changes the disparity in a way that it moves towards 0.

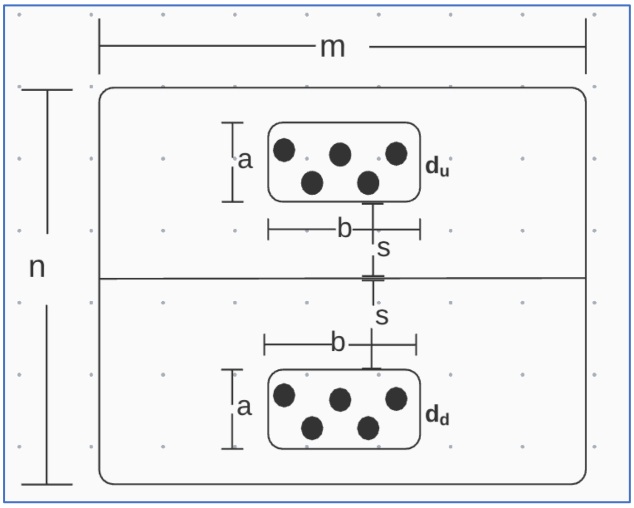

Figure 4. The depth frame, the smaller $a \times b$ rectangles are equidistant from the centre axis.

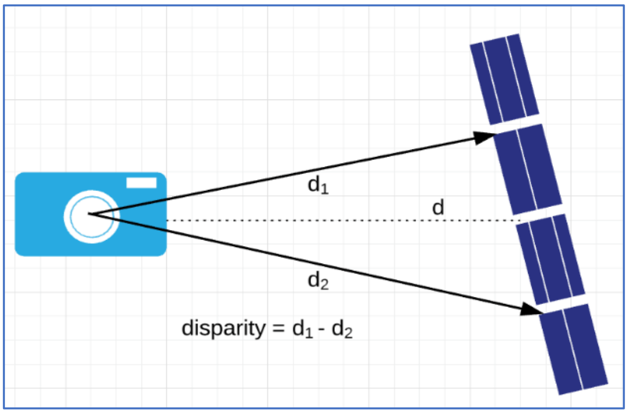

Figure 5. General description: d1 and d2 are the mean distance for each of the rectangles.

4.2 *Collecting data from the sensors*

After moving the camera to the appropriate distance from the wall, we wait for 2 seconds for the exposure to stabilise and then collect 50 frames, these frames are then written to files in .yaml format along with the timestamp of the measurement, the central distance of the measurement, and the computed disparity for the frame.

5 COMPARISON METHODOLOGIES

The techniques we use to perform the qualitative analysis on the frames are:

A. In order to characterise the cameras according to their depth frame quality and to co-relate the data with the external configurations tested, we compute the per-pixel deviation of the depth frame by first computing the gradient of the frame and filtering out the pixels where the gradient is higher than a threshold, marking these pixels as outliers and removing them.

B. We then fit a plane to the depth image of the flat surface (an approximation for wall) and measure the deviations (Figure 7) of the depth values from this plane (Algorithm 1). For each sample, we observe the deviation as the distance from the surface varies. The fitted plane acts (Figure 6) as the ground truth as the distance from the wall is a known value.

Algorithm 1: Computing the error in each depth frame

Input: $\boldsymbol{D}_{mxn}$, a depth frame of dimensions $m \times n$, actual distance (*dist*) at which the frame was taken

$\boldsymbol{D}_{pxq}$ = *crop*($\boldsymbol{D}_{mxn}$, *p, q*)

$\boldsymbol{D}_{pxq}$ = *remove_outliers*($\boldsymbol{D}_{pxq}$)

Define $\boldsymbol{P}$ as a $n \times 3$ matrix to store points

foreach i, j in 0: *p*-1, 0: *q*-1

  [i, j, **D**(*i,j*)] $\xrightarrow{append\ row}$ $\boldsymbol{P}$

end foreach

$\overrightarrow{P_{mean}}$ = *mean_along_each_column*(***P***)

foreach row in **P**

  P[*index, :*] = ***P***[*index, :*] - $\overrightarrow{P_{mean}}$

end foreach

$\boldsymbol{P}_{eigen}$ = $compute\_eigenvector(\boldsymbol{P}' * \boldsymbol{P})$, $\boldsymbol{P}_{eigen}$ is a vector normal to the imaging plane.

Define $\vec{Q}$ = (p/2, q/2, *dist*) as the center point of the actual plane

compute the per pixel deviation matrix **R** as

$\boldsymbol{R}_i = dot(\boldsymbol{P}_i - \overrightarrow{P_{mean}}, \boldsymbol{P}_{eigen})$

mean_error = *mean*($\boldsymbol{R}_i$)

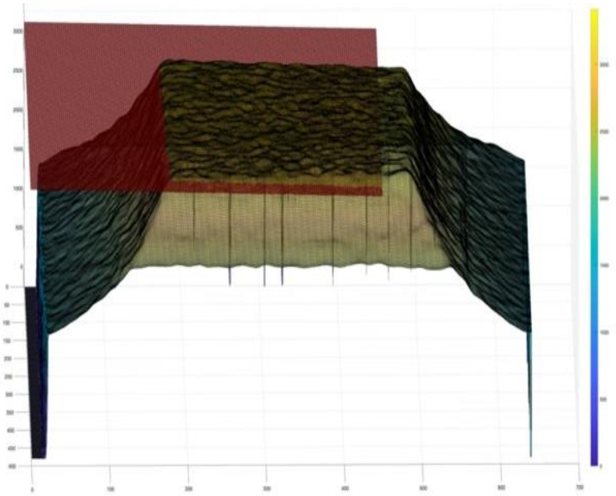

Figure 6. Plane fitted to the cropped flat region, ground truth assumption for the wall.

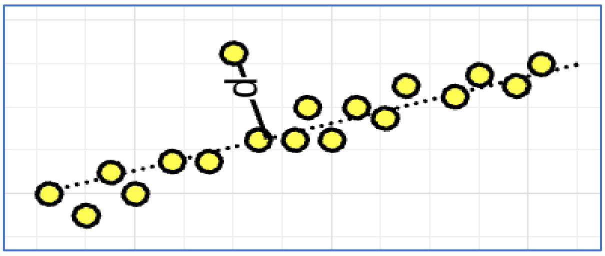

Figure 7. Plane and depth deviations, the dotted line represents the wall while the yellow dots represent the depth measurements.

6 RESULTS AND OBSERVATIONS

The plots of the colour that showed maximum error (deviation) in a particular pixel region of the depth images across all measured distances are shown in Figure 8 (RealSense), Figure 9 (Xtion 2) and Figure 10 (Kinect 2). We observe for the RealSense frame that the colours are equally spread while for the Xtion camera, there is a majority of black pixels which informs us about Xtion's poor performance with black objects. The same performance hit is also observed with Microsoft Kinect 2 with a black surface.

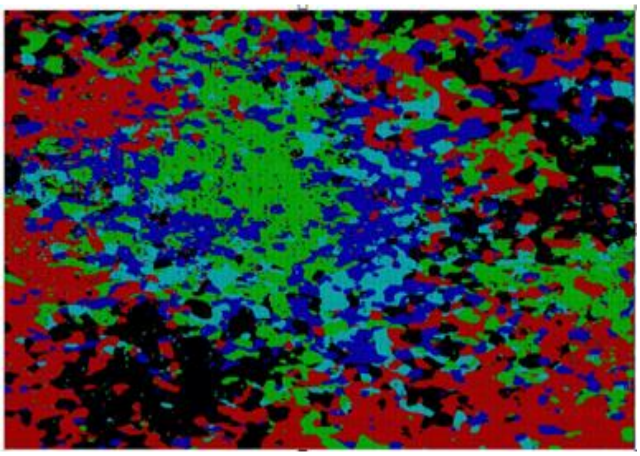

Figure 8. RealSense colour deviation.

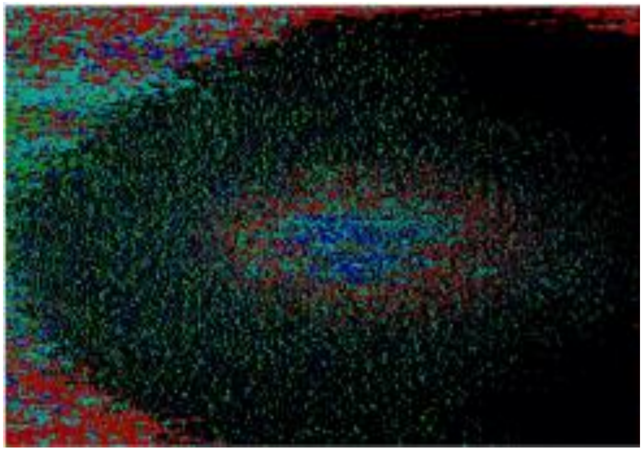

Figure 9. Xtion colour deviation.

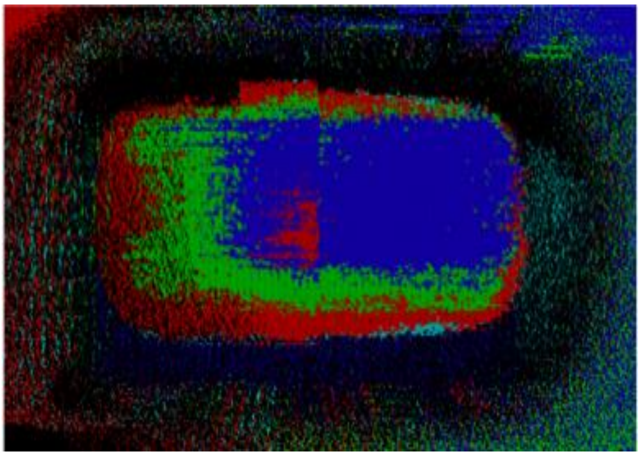

Figure 10. Kinect colour deviation.

Percent outlier pixels for Xtion and Kinect 2 camera for every distance and colour are shown in Table 2. and Table 4. The number of outliers is computed by filtering the pixels with 0 as its depth value in the frame and by computing the gradient of the depth image then filtering the pixels with a high gradient (corresponds to an out of place pixel)

Percent outliers for RealSense camera are shown in Table 3. Notice the high error rates observed for Xtion 2 and Kinect 2 with the black surface.

Table 2. Error pixels per distance, Asus Xtion 2					
Dist./Clr.	RED	BLUE	GREEN	BLACK	WHITE
600	0.65%	0.40%	0.23%	14.92%	0.54%
700	3.01%	1.91%	1.93%	20.34%	1.05%
800	0.24%	0.60%	2.98%	24.92%	3.02%
900	3.02%	2.97%	2.63%	30.00%	1.90%
1000	1.89%	1.35%	0.75%	37.73%	0.79%
1100	2.98%	2.90%	2.83%	37.21%	2.88%
1200	2.76%	2.75%	2.73%	40.20%	2.74%
1300	2.70%	2.65%	2.69%	46.14%	2.76%
1400	2.70%	2.64%	2.63%	52.34%	2.69%
1500	2.62%	2.61%	2.67%	56.68%	2.66%

Table 3. Percentage error pixels per distance, RealSense D435					
Dist./Clr.	RED	BLUE	GREEN	BLACK	WHITE
600	0.29%	0.40%	0.28%	0.31%	0.28%
700	0.33%	0.39%	0.26%	0.37%	0.26%
800	0.29%	0.22%	0.27%	0.29%	0.20%
900	0.13%	0.17%	0.18%	0.21%	0.22%
1000	0.36%	0.53%	0.40%	0.39%	0.32%
1100	0.37%	0.57%	0.49%	0.29%	0.39%
1200	0.12%	0.15%	0.12%	0.23%	0.09%
1300	0.28%	0.55%	0.35%	0.24%	0.21%
1400	0.35%	0.62%	0.47%	0.26%	0.29%
1500	0.32%	0.52%	0.51%	0.12%	0.24%

Table 4. Percentage error pixels per distance, Microsoft Kinect 2					
Dist./Clr.	RED	BLUE	GREEN	BLACK	WHITE
600	26.20%	13.26%	20.28%	10.07%	26.13%
700	26.28%	14.56%	20.22%	8.53%	26.19%
800	2.87%	0.55%	0.59%	8.41%	0.55%
900	0.55%	0.55%	0.55%	7.90%	0.55%
1000	0.55%	0.57%	0.55%	10.74%	0.55%
1100	0.56%	0.63%	0.58%	10.83%	0.60%
1200	0.62%	0.76%	0.71%	12.69%	0.85%
1300	0.85%	0.86%	0.93%	15.63%	0.85%
1400	1.34%	0.79%	1.12%	18.98%	1.30%
1500	1.53%	1.19%	1.07%	21.91%	1.22%

The effect of the colour of surface on the error at every distance between the imaged object and the camera is shown in Figure 11, Figure 12 and Figure 13 for Xtion2, Realsense and Kinect 2 respectively. We observe the error increase as the distance from the sensor increases, although the error is high in the beginning as the imaged surface is too close to the camera. Xtion 2 and Kinect 2 fail to image the black surface properly, and thus the error is very high for most distances. We observe that colour has no effect on the performance of depth cameras as the deviation in Xtion's plots are within experimental error bounds.

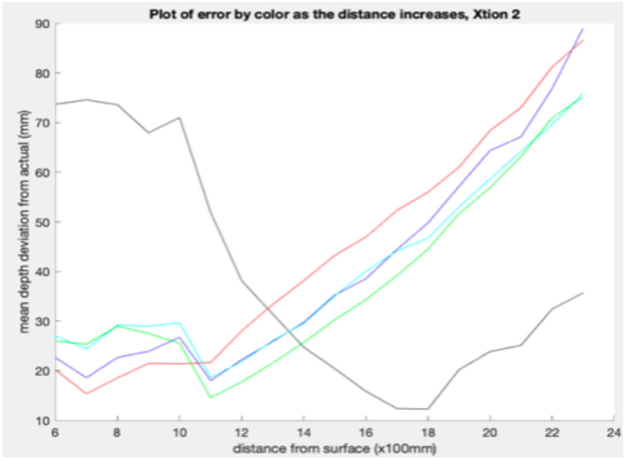

Figure 11. Deviation in mm by error per colour, Xtion2.

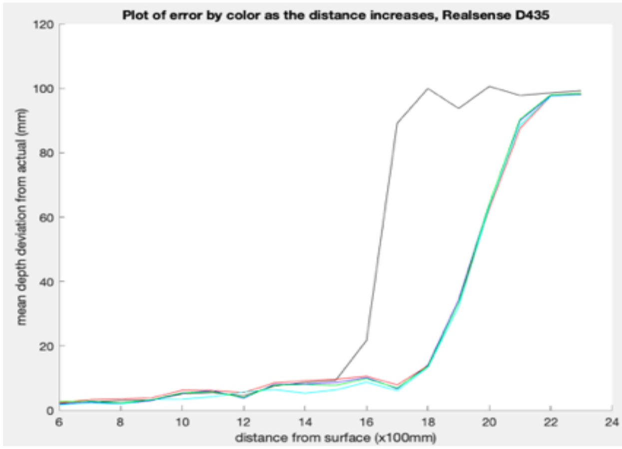

Figure 12. Deviation in mm by error per colour, RealSense.

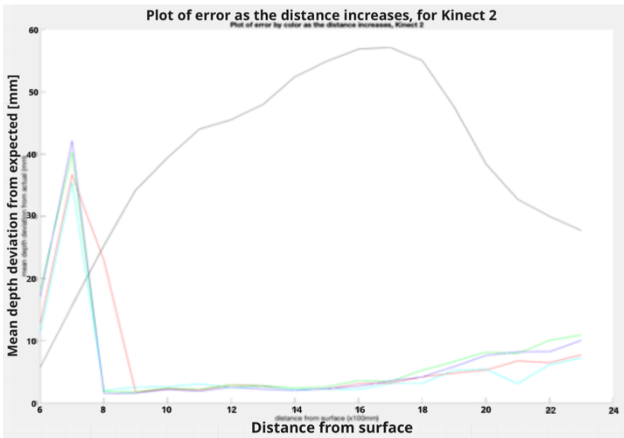

Figure 13. Deviation in mm by error per colour for Kinect.

(X axis: mean depth deviation from actual in mm, Y axis: distance from imaged surface in decimetres).

The histogram of depth pixel values for the distances of 1600mm, 1100mm, 600mm for the depth images are shown in figure 14-22. Figure 14, Figure 15, Figure 16 are for Asus Xtion 2. Figure 17, Figure 18, Figure 19 are for Intel RealSense D435 camera, whereas Figure 20, Figure 21, Figure 22 shows the same for Kinect V2. Xtion's histograms have a higher number of pixels in the distance region different from the actual measurement which signifies more noise. These histograms were generated from frames while imaging a black surface.

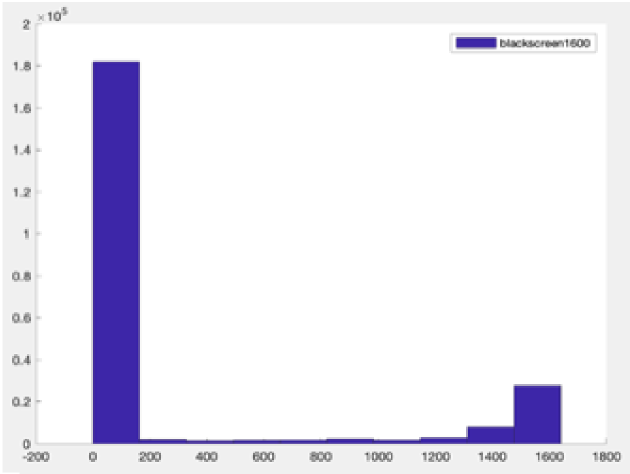

Figure 14. Histogram of the pixel distances for depth images from Asus Xtion, Black screen at 1600 mm away from the wall.

X axis: distance in mm, Y axis: number of pixels at that distance in the depth image

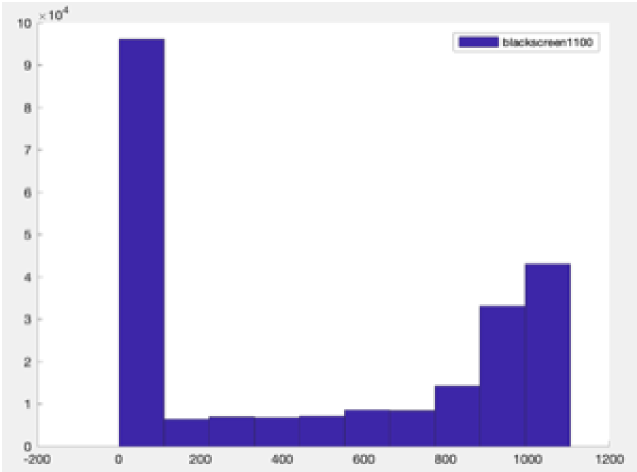

Figure 15. Histogram of the pixel distances for depth images from Asus Xtion, Black screen at 1100 mm away from the wall.

X axis: distance in mm, Y axis: number of pixels at that distance in the depth image

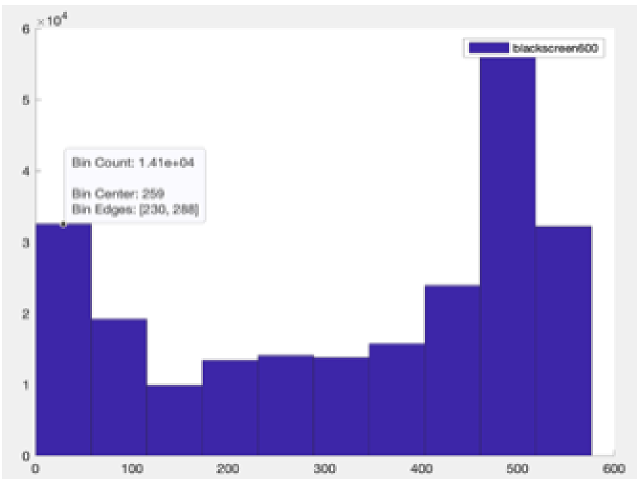

Figure 16. Histogram of the pixel distances for depth images from Asus Xtion, Black screen at 600 mm away from the wall.

X axis: distance in mm, Y axis: number of pixels at that distance in the depth image

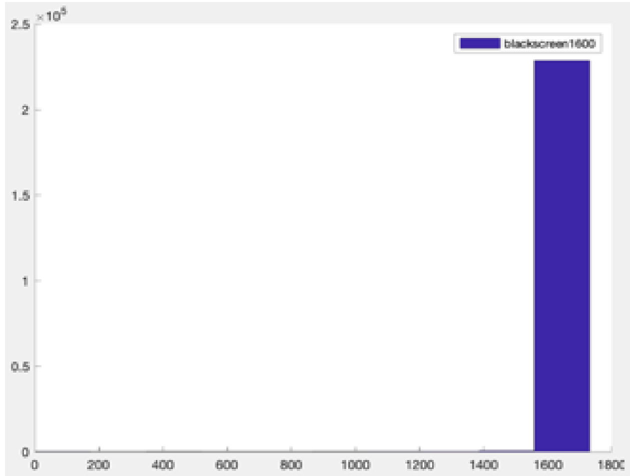

Figure 17. Pixel count histograms for depth images from Intel RealSense, Black screen at 1600 mm away from the wall.

X axis: distance in mm, Y axis: number of pixels at that distance in the depth image

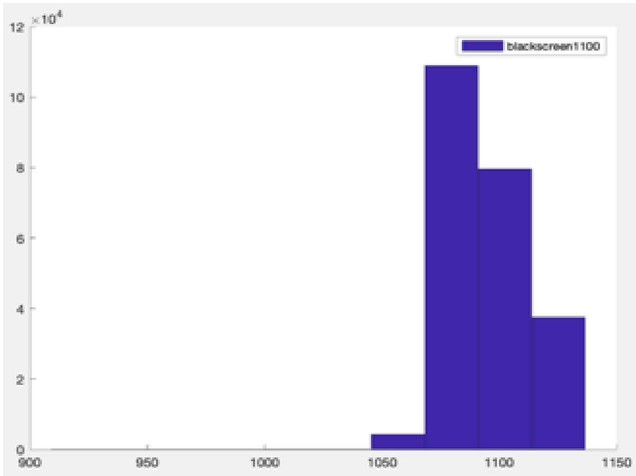

Figure 18. Pixel count histograms for depth images from Intel RealSense, Black screen at 1100 mm away from the wall.

X axis: distance in mm, Y axis: number of pixels at that distance in the depth image

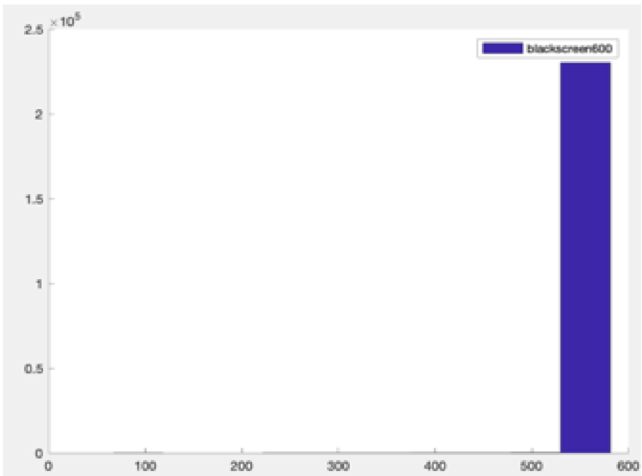

Figure 19. Pixel count histograms for depth images from Intel RealSense, Black screen at 600 mm away from the wall.

X axis: distance in mm, Y axis: number of pixels at that distance in the depth image

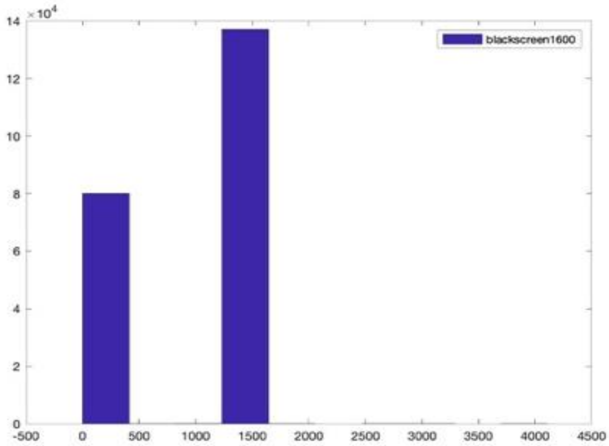

Figure 20. Histogram of the pixel distances for depth images from Kinect 2, Black screen at 1600 mm away from the wall.

X axis: distance in mm, Y axis: number of pixels at that distance in the depth image

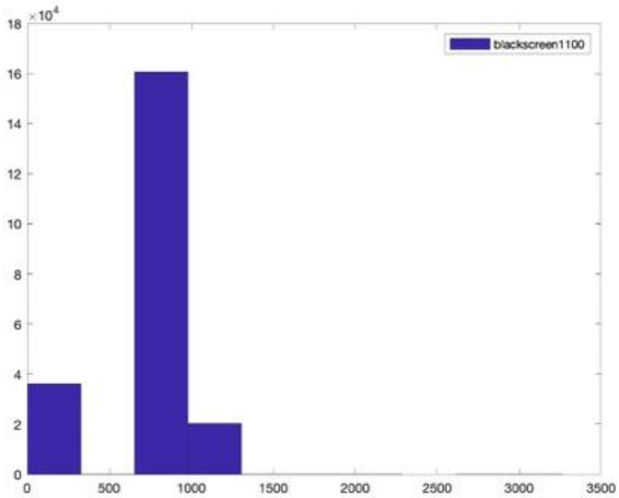

Figure 21. Histogram of the pixel distances for depth images from Kinect 2, Black screen at 1100 mm away from the wall.

X axis: distance in mm, Y axis: number of pixels at that distance in the depth image

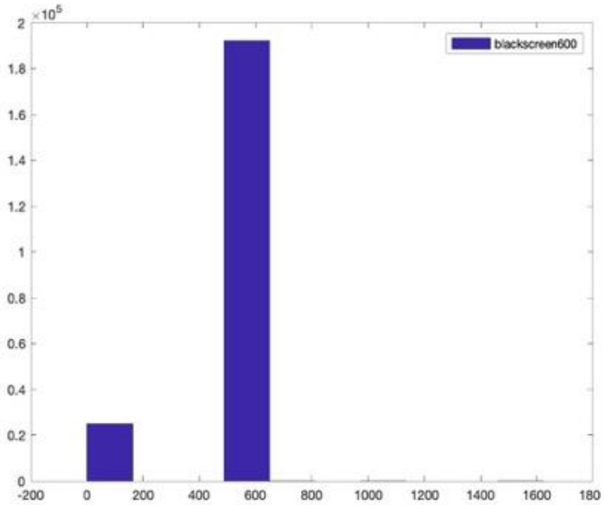

Figure 22. Histogram of the pixel distances for depth images from Kinect 2, Black screen at 600 mm away from the wall.

X axis: distance in mm, Y axis: number of pixels at that distance in the depth image

Xtion 2 fails to image black objects correctly, as is shown in **Figure 23**. Asus Xtion 2 performance with a black surface, ideally this frame should have been a flat surface at 1465mm.

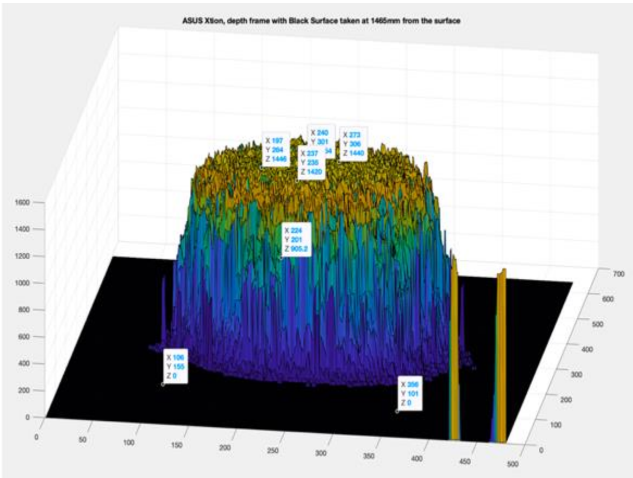

Figure 23. Example depth frame for Xtion with a black imaging target.

7 CONCLUSION

In this paper three popular consumer depth cameras were compared, and the effect of distance from the object being imaged and its colour were measured. We observed that the colour of the surface has a minimum impact with the depth image quality which is in line with the fact that the CMOS sensor in the camera used for the depth images is monochrome in nature; however, we observed that the depth quality decreases as the surface colour tends to black/dark and is better for lighter surfaces which is because the darker color absorbs the incident light from the camera, which in the case of time-of-flight sensors is the laser pulse and for structured light sensor is the output from the projector. We also observed that the depth quality decreases as we go away from the imaged surface which is because the density of points for which the depth data is acquired by the sensor decreases as the distance increases. In future work, we would like to extend our experiment with more sensors, and by varying the texture and reflectance of the imaged surface. We would also like to try different imaging angles and measure its effects on the depth frames.

ACKNOWLEDGMENTS

This work was supported by the Research Platform focused on Industry 4.0 and Robotics in Ostrava Agglomeration project, project number CZ.02.1.01/0.0/0.0/17_049/0008425 within the Operational Programme Research, Development and Education. This article has been also supported by specific research project SP2020/141 and financed by the state budget of the Czech Republic.

REFERENCES

[Langmann 2012] Benjamin Langmann, Klaus Hartmann and Otmar Loffeld, Depth Camera Technology comparison and Technology Evaluation, ZESS – Center for Sensor Systems, University of Siegen, Paul-Bonatz-Strasse 9-11, Siegen, Germany

[Böhler 2003] Böhler, W., Bordas Vicent, M., and Marbs, A., Investigating Laser Scanner Accuracy. In The International Archives of Photogrammetry, 2003. Remote Sensing and Spatial Information Sciences, Vol. XXXIV, Part 5/C15, pp. 696-701.

[Nock 2013] Charles A Nock, Olivier Taugourdeau, Sylvain Delagrange, and Christian Messier, Assessing the Potential of Low-Cost 3D Cameras for the Rapid Measurement of Plant Woody Structure: Sensors (PMID: 24287538)

[Chiu 2019] Chuang-Yuan Chiu, Michael Thelwell, Terry Senior, Simon Choppin John Hart, Jon Wheat. Comparison of depth cameras for three-dimensional reconstruction in medicine: Journal of Engineering in medicine.

[Swoboda 2014] Daniel Maximilian Swoboda, A comprehensive characterisation of the XtionPro Depth Sensor, Höhere Technische Bundes Lehr- und Versuchsanstalt Wiener Neustadt, Abteilung Informatik

[Francis 2015] Francis, Sobers & Anavatti, S.G. & Garratt, Matt & Shim, Hyunbgo. (2015). A ToF-Camera as a 3D Vision Sensor for Autonomous Mobile Robotics. International Journal of Advanced Robotic Systems. 12. 1. 10.5772/61348.

[Kazmi 2014] Kazmi, Wajahat & Foix, Sergi & Alenyà, Guillem & Andersen, Hans. (2014). Indoor and outdoor depth imaging of leaves with time-of-flight and stereo vision sensors: Analysis and comparison. ISPRS Journal of Photogrammetry and Remote Sensing. 88. 128–146. 10.1016/j.isprsjprs.2013.11.012.

[Lottner 2008] Lottner, Oliver & Weihs, Wolfgang & Hartmann, Klaus. (2008). Time-of-flight cameras with multiple distributed illumination units. 40-45. In Proceedings of the 8th conference on Signal processing 2008. computational geometry and artificial vision, Rhodes, Greece, pp. 40-45.

[Rapp 2008] Rapp, H. & Frank, Mario & Hamprecht, Fred & Jähne, Bernd. (2008). A theoretical and experimental investigation of the systematic errors and statistical uncertainties of Time-Of-Flight-cameras. IJISTA. 5. 402-413. 10.1504/IJISTA.2008.021303.

[Sweeney 2019] Sweeney, Chris & Izatt, Gregory & Tedrake, Russ. (2019). A Supervised Approach to Predicting Noise in Depth Images. 796-802. 10.1109/ICRA.2019.8793820.

[Tadic 2019] Tadic, Vladimir & Odry, Ákos & Kecskes, Istvan & Burkus, Ervin & Király, Zoltán & Odry, Peter. (2019). Application of Intel RealSense Cameras for Depth Image Generation in Robotics. WSEAS Transactions on Computers. 18. 107-112.

[Wasenmüller 2016] Wasenmüller, Oliver & Stricker, Didier. (2016). Comparison of Kinect V1 and V2 Depth Images in Terms of Accuracy and Precision. 10.1007/978-3-319-54427-4_3.

[He 2018] Yu He and Shengyong Chen, Advances in sensing and processing methods for three-dimensional robot vision, International Journal of Advanced Robotic Systems March-April 2018

[Bobovsky 2018] Zdenko Bobovský, Václav Krys, Vladimír Mostýn, Kinect v2 infrared images correction. International Journal of Advanced Robotic Systems, Volume: 15 issue: 1

CONTACTS:

Bc. Vyomkesh Kumar Jha

VSB – Technical University of Ostrava

Faculty of Mechanical Engineering

Deptartment of Robotics, Ostrava

17. listopadu 2172/15

708 00 Ostrava

Czech Republic

vyomkesh.kumar.jha@vsb.cz

http://robot.vsb.cz/en