

# Intel RealSense SR305, D415 and L515: Experimental Evaluation and Comparison of Depth Estimation

Francisco Lourenço<sup>1</sup> <sup>a</sup>, Helder Araujo<sup>1</sup> <sup>b</sup>

<sup>1</sup>*Institute of Systems and Robotics, Department of Elec. and Comp. Eng., University of Coimbra, Coimbra, Portugal*  
*francisco.rlourenco@isr.uc.pt, helder@isr.uc.pt*

Keywords: RGB-D cameras, Experimental Evaluation, Experimental Comparison

Abstract: In the last few years Intel has launched several low cost RGB-D cameras. Three of these cameras are the SR305, the L415 and the L515. These three cameras are based on different operating principles. The SR305 is based on structured light projection, the D415 is based on stereo based using also the projection of random dots and the L515 is based on LIDAR. In addition they all provide RGB images. In this paper we perform an experimental analysis and comparison of the depth estimation by the three cameras.

## 1 INTRODUCTION

Consumer level RGB-D cameras are affordable, small and portable. These are some of the main features that make these types of sensors very suitable tools for research and industrial applications, ranging from practical applications such as 3D reconstruction, 6D pose estimation, augmented reality and many more (Zollhöfer et al., 2018). For many applications it is important to know how accurate and precise an RGB-D camera is, in order to understand which sensor best suits the specific application (Cao et al., 2018). With this paper we aim to compare three specific models of RGB-D cameras from Intel, which can be useful for many users and applications.


The sensors are the RealSense SR305, D415 and L515. Each sensor uses different methods to calculate depth. The SR305 uses coded light, where a known pattern is projected into the scene and, by evaluating how this pattern deforms, depth information is computed. The D415 uses stereo vision technology, capturing the scene with two imagers and, by computing the disparity on the two images, depth can be retrieved. Finally, the L515 that measures time-of-flight, i.e., this sensor calculates depth by measuring the delay in between light emission and light reception.


Several different approaches can be used to evaluate depth sensors (P. Rosin, 2019), (A. Fossati, 2013). In this case we focused on the accuracy and repeatability.

For that purpose the cameras were evaluated using depth images of 3D planes at several distances, but whose ground truth position and orientation distance were not used. In addition also the average number of depth points per image that the cameras failed to calculate depth (and their standard deviation) were also used. The number of outliers per image (points for which the depth was outside an interval). Accuracy was measured in terms of point-to-plane distance and precision was measured as the repeatability of 3D model reconstruction, i.e., standard deviation of the parameters of the estimated 3D model (in this case a 3D plane). Moreover, we employ directional statistics (Mardia and Jupp, 1999) on the normal vectors of the planes in order to better illustrate how this models variate.

## 2 RELATED WORK

Depth cameras and RGB-D cameras have been analyzed and compared in a number of different ways. In (Halmetschlager-Funek et al., 2019) several parameters of ten depth cameras were experimentally analyzed and compared. In addition to an analysis of the response of the cameras to different materials, noise characteristics and precision were also evaluated. In (Chuang-Yuan Chiu and Wheat, 2019) depth cameras were compared considering medical applications and their specific requirements. Another comparison for medical applications is performed in (Siena et al., 2018). A comparison for agricultural applications is

<sup>a</sup>  <https://orcid.org/0000-0001-7081-5641>

<sup>b</sup>  <https://orcid.org/0000-0002-9544-424X>

performed in (Vit and Shani, 2018). Analysis for robotic applications is performed in (Changjuan Jing, 2017). In (Anxionnat et al., 2018) several RGB-D sensors are analyzed and compared based on controlled displacements, with precision and accuracy evaluations.

### 3 METHODOLOGY

#### 3.1 Materials

As aforementioned, the sensors used in this evaluation employ different depth estimation principles, which not only yields information about the sensors performance, but also on how these technologies compare to each other (for the specific criteria used).

The SR305 uses coded light, the D415 uses stereo vision and the L515 uses LIDAR. The camera specifications are represented on table 1. The three cameras were mounted on the standard tripod.

To ensure constant illumination conditions a LED ring (eff, 2020) was used as the only room illumination source.

Table 1: Sensors resolution (px\*px) and range (m).

Sensor	SR305	D415	L515
Depth	640x480	1280x720	1024x768
Color	1920x1080	1920x1080	1920x1080
Range	[0.2 1.5]	[0.3 10]	[0.25 9]

Note that the values on table 1 are upper bounds, meaning that the specifications may vary with different configurations of the sensors. It is also important to mention that the D415 range may vary with the light conditions.

#### 3.2 Experimental setup

Each camera was mounted on a tripod and placed at a distance  $d$  of a wall. The wall is white and covers all the field of view of the cameras. The optical axes of the sensors are approximately perpendicular to the wall. Placed above of the camera is the light source, above described. The light source points to the wall with the same direction as the camera. For practical reasons the light source is slightly behind the camera in such way that the camera does not interfere with the light. A laptop is placed behind the camera where the camera software is executed and where the images are stored. All the experiments took place at night, avoiding any unwanted daylight. Hence, the light on the room was kept constant in between experiments and always sourced by the same element.

Camera, light source and laptop were placed on top of a structure. The reason why we wanted everything to be high relative to the ground was to ensure that neither floor or ceiling were captured by the sensors. For each distance at which the cameras were placed, 100 images were acquired. The distances for which both D415 and L515 were tested are 0.5m, 0.75m, 1m, 1.25m, 1.5m, 1.75m, 2m, 2.5m, 3m and 3.5m. The furthest distance was the maximum distance for which neither floor or ceiling appeared on the images. The SR305 was tested at 0.3m, 0.4m, 0.5m, 0.75m, 1m, 1.25m and 1.5m. In this case, the furthest distance is the maximum specified range for the SR305 sensor.

The experiments started at the closest distance. The sensors were switched right after the other sequentially and, after all the images were obtained at that distance, all the structure was moved away from the wall by a distance of the aforementioned intervals. The structure moved approximately perpendicularly to the wall.

For the D415 and the L515 sensors we used custom configurations. For the SR305 we used default configuration. These configurations were exactly the same as those of table 1.

In figure 1 the experimental setup is illustrated.

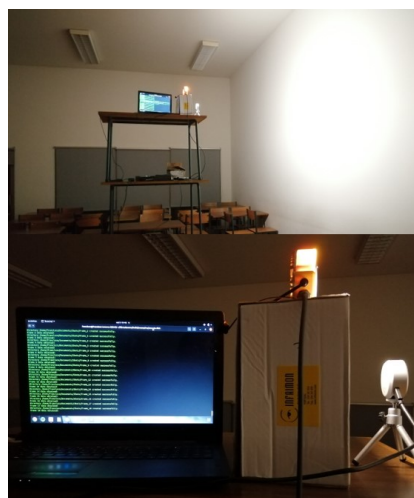


Figure 1: Experimental Setup.

#### 3.3 Software

To deal with the sensors, the Intel RealSense SDK 2.0 was used. The Intel RealSense Viewer API was used to check the sensors behavior right before each execution, to check for the direction of the optical axis and distance. All the other tasks were executed using custom code and the librealSense2 library. These tasks

include both image acquisition and storing, camera configuration and point cloud generation. This part of the work was executed using Ubuntu. All the statistical evaluation was performed in MatLab, on Windows 10.

### 3.4 Experimental Evaluation

#### 3.4.1 Performance

The performance of the sensors was measured in two ways. First, we calculated the average number of points for which the sensor failed to measure depth and also the standard deviation of the same number of points. Then we do the same for outliers.

Whenever the Intel RealSense SDK 2.0 and camera fail to measure depth at some point, the corresponding depth is defined as zero. Hence, all we do here is to count pixels in the depth image with depth equal to zero.

Depth values also contain outliers. Outliers can be defined in several ways. In this case we considered as an outlier every point with a depth value differing 10cm from the expected distance, given the specific geometric configuration and setup.

As described in the Intel RealSense D415 product DataSheet (D41, 2020), the D415 sensor has an invalid depth band, which is a region in the depth image for which depth can not be computed.

The coordinate system of the left camera is used as the reference coordinate system for the stereo camera. The left and right cameras have the same field of view, but due to their relative displacement there is an area in the left image for which it is no possible to compute disparities, since the corresponding 3D volume is not visible in the right camera. Resulting in a non-overlap region of the left and right cameras for which it is not possible to measure depth. This region appears in the leftmost area of the image and is illustrated in figure 2. The total number of pixels in the invalid depth band can be calculated in pixels as follows:

$$InvalidDepthBand = \frac{VRES * HRES * B}{2 * Z * \tan(\frac{HFOV}{2})} \quad (1)$$

Where  $VRES$  and  $HRES$  stand for vertical and horizontal resolution respectively (720px and 1280px),  $B$  is the baseline  $\rightarrow$  55mm,  $Z$  is the distance of scene from the depth module  $\rightarrow$   $d$  and  $HFOV$  is the horizontal field of view  $\rightarrow$   $64^\circ$ .

Bearing that in mind, the pixels in the invalid depth band were ignored in our calculations.

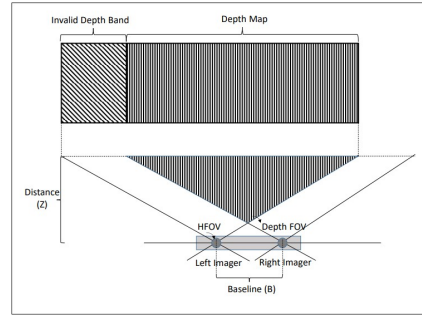


Figure 2: Invalid Depth Band.

#### 3.4.2 Plane Fitting

Point clouds were first obtained by using the depth data, the image pixel coordinates and the camera intrinsics. This is possible because we have depth information, letting the coordinate  $z$  be equal to the measured depth at that point, i.e.,  $z = d_m$  the following equations can be applied:

$$x = z * \frac{u - pp_x}{f_x} \quad (2) \quad y = z * \frac{v - pp_y}{f_y} \quad (3)$$

Where  $(u, v)$  are the pixel coordinates,  $(pp_x, pp_y)$  are the coordinates of the principal point,  $f_x$  and  $f_y$  are the focal lengths in pixel units.

The point clouds correspond to a wall, thus it is possible to fit a plane to the data.

Since we handle ourselves the outliers, we performed the estimation of the plane equation using standard least-squares regression, by means of the singular value decomposition, instead of robust approaches such as RANSAC.

The model we want to be regressed to the point clouds is the general form of the plane equation:

$$x * n_x + y * n_y + z * n_z - d = 0 \quad (4)$$

Where  $(n_x, n_y, n_z)$  stands for the unit normal, and  $d$  stands for distance from the plane to the origin.

If we now build a  $n * 4$  matrix from the point clouds with  $n$  points, which we will denote as matrix  $P$ . We can rewrite equation 4:

$$0 = \underbrace{\begin{bmatrix} x_1 & y_1 & z_1 & -1 \\ x_2 & y_2 & z_2 & -1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & -1 \end{bmatrix}}_P \begin{bmatrix} n_x \\ n_y \\ n_z \\ d \end{bmatrix} \quad (5)$$

By computing the singular value decomposition on matrix  $P$  as:

$$P = U * \Sigma * V' \quad (6)$$

We can now use the values of the column of matrix  $V$  that corresponds to the smallest eigenvalue in matrix  $\Sigma$ , as the parameters  $n_x^*, n_y^*, n_z^*$  and  $d^*$  of the

plane that fit that point cloud. Then, we normalize the normal vector of the plane, which will become handy in further calculations and recover the true distance in meters of the plane from the sensor.

### 3.4.3 Accuracy

For the accuracy analysis, we compute the point-to-plane distance. For each point of the point cloud, we use the fitted plane equation to compute the errors, i.e, the point-to-plane distance. We then calculate the average root mean square error for each set of 100 images as an accuracy measurement. For the sake of comparison, we perform this computation for two different thresholds values for outlier rejection ( $\pm 10cm$  and  $\pm 35cm$ ), and another one where we use all points with the measured depth.

### 3.4.4 Precision

In this work, precision is measured as per image plane consistency, i.e., how the plane model changes in between images of the same sensor at the same distance. As neither the scene or the sensor change while taking the pictures, we could expect the models to be the exact same if we had an ideal sensor. Thus, by measuring the the standard deviation of the plane model parameters in between images, we might be able to better understand how consistent the sensors are with their measurements, as well as how this consistency varies with the distance.

Additionally, we also transform the normal vector of the plane into spherical coordinates, where we can perform analysis of directional statistics as all the normals are distributed on a spherical surface. Specifically, the circular mean and standard deviation of angles  $\theta$  and  $\phi$ , and the spherical variance of the normal vectors. Since  $\vec{n}_i$  is unitary, its norm  $\rho$  is 1.

Let  $\theta$  and  $\phi$  be the azimuth and altitude angles of  $\vec{n}_i$ :

$$\theta_i = \arctan \frac{n_{y_i}}{n_{x_i}} \quad (7) \quad \phi_i = \arctan \frac{n_{z_i}}{\sqrt{n_{x_i}^2 + n_{y_i}^2}} \quad (8)$$

As in (Mardia and Jupp, 1999), the circular mean of the angles above can be computed as follows:

$$\bar{\theta} = \arctan \frac{\sum_{i=1}^n \sin \theta_i}{\sum_{i=1}^n \cos \theta_i} \quad (9)$$

$$\bar{\phi} = \arctan \frac{\sum_{i=1}^n \sin \phi_i}{\sum_{i=1}^n \cos \phi_i} \quad (10)$$

Now, in order to show how the spherical variance is computed, we need to introduce vector  $\vec{n}$ , which is the vector whose components are the mean of each component of  $\vec{n}_i$ . If we now compute the norm of  $\vec{n}$  and call it  $\bar{R}$ , the spherical variance is calculated as follows:

$$V = 1 - \bar{R} \quad (11)$$

## 4 Results

### 4.1 Performance analysis

In tables 2, 3 (In Appendix) and figure 3, we show the results for the average number of failed points and the standard deviation of the number of failed points per image (points where the sensor could not compute depth). As it can be verified, the L515 sensor outperforms both D415 and SR305. This sensor not only showed to be capable of estimating more depth data relatively to its resolution, but the results also show that that number (of failed points) remains almost constant up until 2 meters of distance. This can be explained by the fact that this sensor uses LIDAR technology, which is more robust than the stereo vision and coded light approaches, since LIDAR directly computes depth.

On the other hand, the D415 sensor shows much better results than the SR305.

A relevant detail of this performance measure is the fact that the standard deviation of the number of failed depth points for the D415 is not strongly dependent on the distance, whereas that does not seem to be the case for the L515.

Tables 4, 5 (In Appendix) and figure 4 include the results for the number of outliers. Just as it happened in terms of the average number of failed depth points, the data for the D415 camera show an increase on the average number of outliers as the distance increases. On the other hand, the number of outliers for the SR305 seems to decrease with increasing distances. These results are quite different from those presented in table 2 (In Appendix). The main reason for these results is probably due to the fact that the SR305 camera is estimating a relatively small number of depth points at higher distances, therefore decreasing the probability of occurrence of outliers.

To better illustrate this, we show on figure 5 a sample image taken with the SR305 sensor at 1.5 meters of distance, where it is notorious the small amount of points for which this sensor is measuring depth (the dark region represents points where depth was not computed).

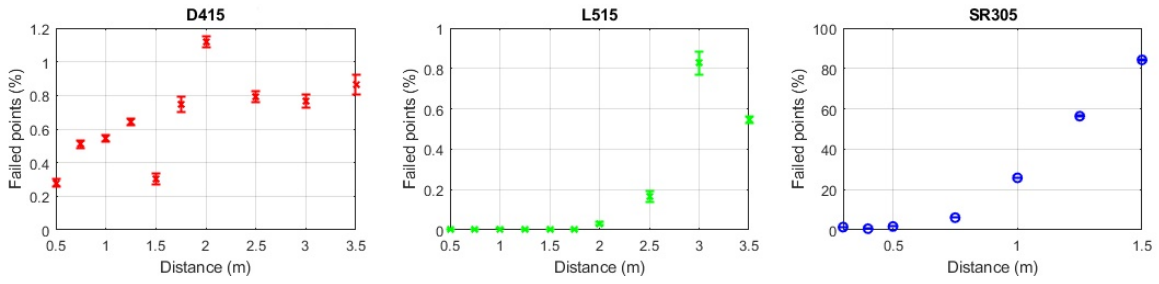


Figure 3: Failed Points: D415 vs L515 vs SR305.

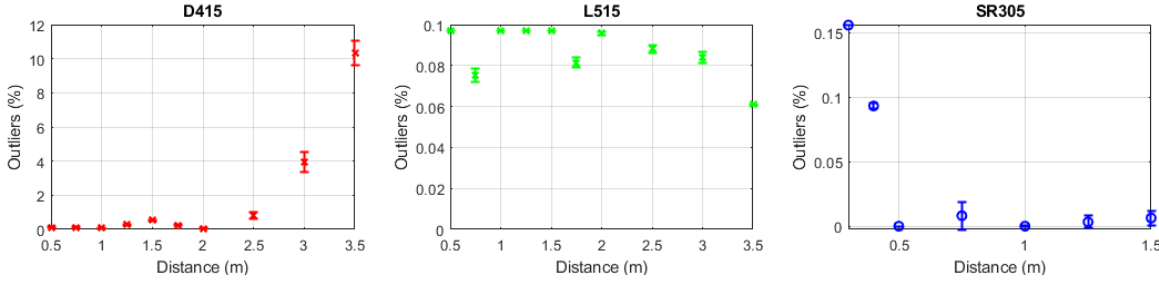


Figure 4: Outliers: D415 vs L515 vs SR305.

On the other hand, for the case of the L515, the number of outliers is essentially independent of the distance. Even though there is some fluctuation on the numbers, the variation is relatively small. Considering the standard deviation of the number of outliers per image obtained with the L515 at 0.75, 1.5 and 1.75 meters, we can see that it is zero, meaning that the total number of outliers per image stayed constant over the 100 images. This led us to determine the pixels where L515 generated outliers. We found out that the miscalculated points always correspond to the same pixels from the leftmost column of the image. This may be due to an issue of the camera used in the experiments, which requires further investigation.

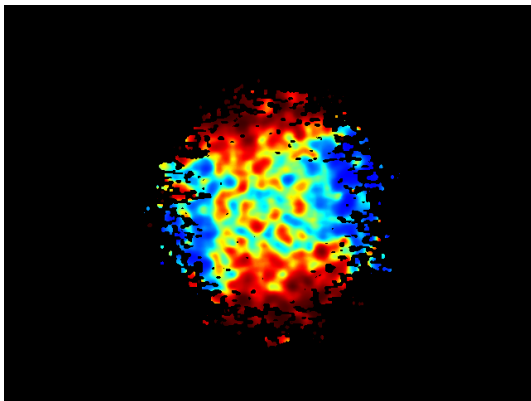


Figure 5: SR305 depth image at 1,5m.

## 4.2 Accuracy analysis

The results of the accuracy study are represented on table 6 (In Appendix) and in figure 6, where we plot the point-to-plane RMSE distances for the three cameras, taking into account only points whose distances from the expected distance are within 10cm.

Again, the sensor that achieves the best results is the L515. It not only has the lowest average root mean square error per image, but it also shows to be the less sensitive to distance. It should be mentioned that the image acquisition conditions for the L515 are close to be optimal since the images were acquired indoors, without daylight illumination. In addition the object was a white wall with good reflectance and low but not inexistent surface roughness.

In the case of camera D415 its RMSE follows a smooth increasing pattern as it goes further from the wall. On the other hand, the RMSE for the SR305 camera does not vary as smoothly with distance. The RMSE values for this sensor increase until 0.75m and then start to decrease until 1m. In fact, 1 meter is the distance for which the SR305 is optimized (SR3, 2020), therefore one should expect this sensor to work better within this range.

## 4.3 Precision analysis

The results show that the camera L515 is significantly more consistent than the other sensors. The results from tables 7 and 8 (In Appendix), show L515 to be more precise in terms of 3D estimation. It is notice-

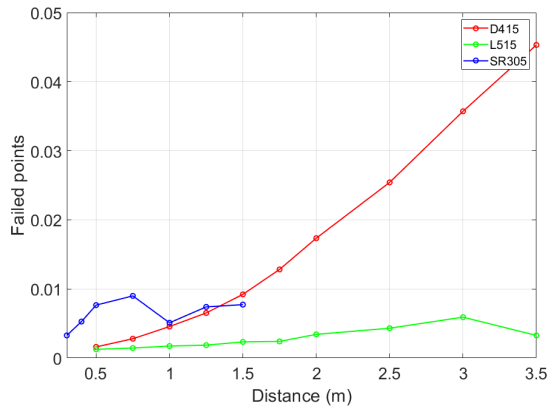


Figure 6: Point-to-plane RMSE D415 vs L515 vs SR305 -  $\pm 10cm$ .

able how this sensor seems to be very consistent in between pictures but also for different distances.

Considering the directional statistics results in table 8 (In Appendix), we can see that the values for the angle  $\theta$  change frequently in a non-systematic way. This is because as  $\phi$  gets closer to  $90^\circ$ , the components  $n_x$  and  $n_y$  of the normal vector get closer to zero, which will lead to large variations of the angle  $\theta$ .

The spherical variation behaves just as expected, showing again that the L515 sensor is the most stable, and that the measurements from the other two are more distance sensitive.

For ease of comprehension of our precision results, we plot on figures 7 and 8 the standard deviation of parameter d of the plane for all cameras at all distances, and also the spherical variation.

## 5 CONCLUSION

In this paper we described a set of experiments performed to compare the depth estimation performance of three RGB-D cameras from Intel, namely the SR305, the D415 and the L515. Overall the results show that the L515 is, in general, more accurate and precise than the other two, while providing also more stable and consistent measurements in the specific environmental conditions of the experiments (indoors with controlled and stable illumination).

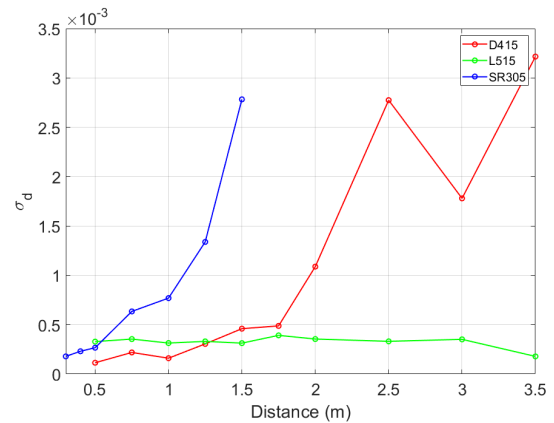


Figure 7: Parameter d standard deviation.

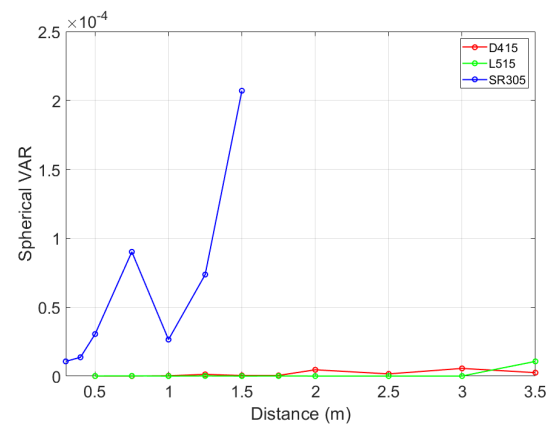


Figure 8: Spherical Variation.

## REFERENCES

- (2020). Configurable and powerful ring lights for machine vision. <https://www.ffmpeg.com/en/products/ring/effi-ring/>. (Accessed: 05-10-2020).
- (2020). Intel® realSense™ d400 series product family. <https://www.intelrealsense.com/depth-camera-d415/>. (Accessed: 01-10-2020).
- (2020). Intel® realSense™ depth camera sr305. <https://www.intelrealsense.com/depth-camera-sr305/>. (Accessed: 02-10-2020).
- A. Fossati, J. Gall, H. G. X. R. K. K., editor (2013). *Consumer Depth Cameras for Computer Vision*. Springer.
- Anxionnat, A., Voros, S., and Troccaz, J. (2018). Comparative study of RGB-D sensors based on controlled displacements of a ground-truth model 1. In *2018 15th International Conference on Control, Automation, Robotics and Vision, ICARCV 2018*, pages 125–128.
- Cao, Y.-P., Kobbelt, L., and Hu, S.-M. (2018). Real-time High-accuracy 3D Reconstruction with Consumer RGB-D Cameras. *ACM Transactions on Graphics*, 1(1).
- Changjuan Jing, Johan Potgieter, F. N. R. W. (2017). A comparison and analysis of rgb-d cameras' depth per-

formance for robotics application. In *Proc. of the Int- Conf. on on Mechatronics and Machine Vision in Practice*.

- Chuang-Yuan Chiu, Michael Thelwell, T. S. S. C. J. H. and Wheat, J. (2019). Comparison of depth cameras for three-dimensional reconstruction in medicine. *Journal of Engineering in Medicine*, 233(9).
- Halmetschlager-Funek, G., Suchi, M., Kampel, M., and Vincze, M. (2019). An empirical evaluation of ten depth cameras: Bias, precision, lateral noise, different lighting conditions and materials, and multiple sensor setups in indoor environments. *IEEE Robotics and Automation Magazine*, 26(1):67–77.
- Mardia, K. V. and Jupp, P. E. (1999). *Directional Statistics*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2nd edition.
- P. Rosin, Yu-Kun Lai, L. S. Y. L., editor (2019). *RGB-D Image Analysis and Processing*. Springer.
- Siena, F. L., Byrom, B., Watts, P., and Breedon, P. (2018). Utilising the Intel RealSense Camera for Measuring Health Outcomes in Clinical Research. *Journal of Medical Systems*, 42(3):1–10.
- Vit, A. and Shani, G. (2018). Comparing RGB-D sensors for close range outdoor agricultural phenotyping. *Sensors (Switzerland)*, 18(12):1–17.
- Zollhöfer, M., Stotko, P., Görlitz, A., Theobalt, C., Nießner, M., Klein, R., and Kolb, A. (2018). State of the art on 3D reconstruction with RGB-D cameras. *Computer Graphics Forum*, 37(2):625–652.

## APPENDIX

Table 2: Average of failed points ratio per image by distance in meters.

d	D415	L515	d	SR305
0.5	0.2801%	0.0020%	0.3	1.3967%
0.75	0.5099%	0.0001%	0.4	0.6062%
1	0.5414%	0.0023%	0.5	1.6930%
1.25	0.6415%	0.0001%	0.75	6.0734%
1.5	0.3025%	0.0014%	1	25.8181%
1.75	0.7472%	0.0017%	1.25	56.3898%
2	1.1202%	0.0320%	1.5	84.2698%
2.5	0.7945%	0.1648%	—	—
3	0.7660%	0.8263%	—	—
3.5	0.8644%	0.5456%	—	—

Table 3: Standard deviation of failed points ratio per image by distance in meters.

d	D415	L515	d	SR305
0.5	0.0248%	0.0023%	0.3	0.0397%
0.75	0.0224%	0.0003%	0.4	0.0198%
1	0.0195%	0.0050%	0.5	0.0415%
1.25	0.0211%	0.0004%	0.75	0.0783%
1.5	0.0329%	0.0033%	1	0.1282%
1.75	0.0458%	0.0025%	1.25	0.1886%
2	0.0319%	0.0100%	1.5	0.2714%
2.5	0.0335%	0.0278%	—	—
3	0.0378%	0.0570%	—	—
3.5	0.0585%	0.0155%	—	—

Table 4: Average of outliers  $\pm 10cm$  ratio per image by distance in meters.

d	D415	L515	d	SR305
0.5	0.1141%	0.0968%	0.3	0.1560%
0.75	0.1112%	0.0753%	0.4	0.0933%
1	0.0667%	0.0968%	0.5	0%
1.25	0.2809%	0.0968%	0.75	0.0081%
1.5	0.5387%	0.0968%	1	3.9062%
1.75	0.2568%	0.0813%	1.25	0.0032%
2	0.0558%	0.0957%	1.5	0.0063%
2.5	0.8361%	0.0882%	—	—
3	3.9436%	0.0839%	—	—
3.5	10.3307%	0.0609%	—	—

Table 5: Standard deviation of outliers  $\pm 10cm$  ratio per image by distance in meters.

d	D415	L515	d	SR305
0.5	0.0209%	0%	0.3	0.0004%
0.75	0.0193%	0.0034%	0.4	0.0018%
1	0.0139%	1.2715%	0.5	0%
1.25	0.0226%	0%	0.75	0.0105%
1.5	0.0380%	0%	1	0.0003%
1.75	0.0431%	0.0025%	1.25	0.0046%
2	0.0168%	0.0008%	1.5	0.0057%
2.5	0.1892%	0.0018%	—	—
3	0.5601%	0.0027%	—	—
3.5	0.7226%	0.0001%	—	—

Table 6: Sensors' accuracy in terms of average root mean square point-to-plane distance error per image.

<i>Cam<sub>dist</sub></i>	$\pm 10cm$	$\pm 35cm$	All Points
<i>D415<sub>0,5m</sub></i>	0,001608	0,004417	0,121739
<i>D415<sub>0,75m</sub></i>	0,002793	0,002793	0,086991
<i>D415<sub>1m</sub></i>	0,004556	0,004556	0,079271
<i>D415<sub>1,25m</sub></i>	0,006525	0,006525	0,036677
<i>D415<sub>1,5m</sub></i>	0,009215	0,009216	0,138918
<i>D415<sub>1,75m</sub></i>	0,012826	0,012836	0,292978
<i>D415<sub>2m</sub></i>	0,017348	0,017410	0,092886
<i>D415<sub>2,5m</sub></i>	0,025423	0,026837	0,035765
<i>D415<sub>3m</sub></i>	0,035715	0,042479	0,068900
<i>D415<sub>3,5m</sub></i>	0,045316	0,062525	0,064792
<i>L515<sub>0,5m</sub></i>	0,000836	0,000836	134,852530
<i>L515<sub>0,75m</sub></i>	0,001263	0,001263	202,285150
<i>L515<sub>1m</sub></i>	0,001446	0,001446	150,912707
<i>L515<sub>1,25m</sub></i>	0,001737	0,001737	7,555782
<i>L515<sub>1,5m</sub></i>	0,001867	0,001867	14,588204
<i>L515<sub>1,75m</sub></i>	0,002335	0,002335	22,389234
<i>L515<sub>2m</sub></i>	0,002408	0,002409	15,541528
<i>L515<sub>2,5m</sub></i>	0,003426	0,003427	14,865477
<i>L515<sub>3m</sub></i>	0,004319	0,004322	25,024772
<i>L515<sub>3,5m</sub></i>	0,005918	0,005931	13,835301
<i>SR305<sub>0,3m</sub></i>	0,003270	0,035584	113,994067
<i>SR305<sub>0,4m</sub></i>	0,005274	0,005274	114,888641
<i>SR305<sub>0,5m</sub></i>	0,007658	0,007658	0,007711
<i>SR305<sub>0,75m</sub></i>	0,008993	0,009049	0,009157
<i>SR305<sub>1m</sub></i>	0,005100	0,005101	0,005516
<i>SR305<sub>1,25m</sub></i>	0,007402	0,007444	0,008199
<i>SR305<sub>1,5m</sub></i>	0,007728	0,008018	0,010544



Table 7: Camera precision in terms of plane modelling consistency.

$Cam_{dist}$	$\bar{n}_x$	$\sigma_{n_x}$	$\bar{n}_y$	$\sigma_{n_y}$	$\bar{n}_z$	$\sigma_{n_z}$	$\bar{d}$	$\sigma_d$
$D415_{0,5m}$	0,003920	0,000250	0,004054	0,000209	0,999984	0,000001	0,497405	0,000114
$D415_{0,75m}$	0,003660	0,000304	-0,008757	0,000195	0,999955	0,000002	0,748806	0,000219
$D415_{1m}$	0,017304	0,000723	-0,001331	0,000257	0,999849	0,000013	1,002985	0,000160
$D415_{1,25m}$	0,014051	0,001554	-0,002677	0,000356	0,999898	0,000018	1,252395	0,000305
$D415_{1,5m}$	0,018492	0,000870	0,003032	0,000429	0,999824	0,000016	1,501446	0,000460
$D415_{1,75m}$	0,021697	0,000846	0,000344	0,000578	0,999765	0,000018	1,756973	0,000488
$D415_{2m}$	0,012880	0,002914	0,002427	0,000864	0,999914	0,000029	2,005980	0,001088
$D415_{2,5m}$	0,009963	0,001546	-0,003214	0,000926	0,999945	0,000014	2,522379	0,002775
$D415_{3m}$	0,013856	0,003168	-0,000582	0,001087	0,999904	0,000033	2,998425	0,001780
$D415_{3,5m}$	0,006606	0,001811	0,001758	0,001337	0,999977	0,000012	3,500792	0,003216
$L515_{0,5m}$	-0,000046	0,000613	-0,002877	0,000230	0,999996	0,000001	0,497359	0,000354
$L515_{0,75m}$	0,001820	0,000460	-0,002682	0,000156	0,999995	0,000001	0,749530	0,000328
$L515_{1m}$	-0,001932	0,000608	-0,005663	0,000154	0,999982	0,000002	1,002775	0,000354
$L515_{1,25m}$	-0,000007	0,000313	0,002108	0,000076	0,999998	0,000000	1,252812	0,000314
$L515_{1,5m}$	-0,003052	0,000522	-0,000919	0,000194	0,999995	0,000002	1,502205	0,000331
$L515_{1,75m}$	0,001811	0,000330	0,000306	0,000100	0,999998	0,000001	1,753154	0,000313
$L515_{2m}$	-0,000804	0,000451	-0,000515	0,000167	1,000000	0,000000	2,005209	0,000392
$L515_{2,5m}$	0,000315	0,000355	-0,000868	0,000142	1,000000	0,000000	2,505247	0,000355
$L515_{3m}$	-0,002107	0,000326	-0,003877	0,000152	0,999990	0,000001	3,001815	0,000331
$L515_{3,5m}$	0,001704	0,000398	-0,006756	0,000083	0,999976	0,000001	3,506949	0,000351
$SR305_{0,3m}$	0,004422	0,004624	-0,005599	0,000103	0,999975	0,000027	0,299631	0,000179
$SR305_{0,4m}$	0,007715	0,005231	-0,000822	0,000087	0,999970	0,000062	0,399063	0,000231
$SR305_{0,5m}$	0,016861	0,007814	0,005249	0,000119	0,999844	0,000156	0,497196	0,000267
$SR305_{0,75m}$	-0,016562	0,013423	-0,000126	0,000283	0,999863	0,000257	0,750328	0,000634
$SR305_{1m}$	-0,016538	0,007281	0,006616	0,000178	0,999841	0,000100	1,002685	0,000769
$SR305_{1,25m}$	-0,026590	0,012128	-0,002321	0,000503	0,999644	0,000255	1,258102	0,001339
$SR305_{1,5m}$	-0,022823	0,020308	-0,002936	0,001147	0,999735	0,000582	1,503880	0,002783

Table 8: Camera precision in terms of plane normal vector angles standard deviation and spherical variance.

$Cam_{dist}$	$\bar{\theta}$	$\sigma_{\theta}$	$\bar{\phi}$	$\sigma_{\phi}$	V
$D415_{0,5m}$	45,980574	1,978900	89,676678	0,014909	0,0000001
$D415_{0,75m}$	-67,318443	1,956108	89,455904	0,009200	0,0000001
$D415_{1m}$	-4,405966	0,877366	89,005474	0,041248	0,0000003
$D415_{1,25m}$	-11,005967	2,653767	89,179798	0,085261	0,0000013
$D415_{1,5m}$	9,331121	1,399447	88,925965	0,049031	0,0000005
$D415_{1,75m}$	0,910185	1,519463	88,756166	0,048559	0,0000005
$D415_{2m}$	11,595530	7,702607	89,246442	0,162710	0,0000046
$D415_{2,5m}$	-18,211600	7,250585	89,396560	0,079415	0,0000016
$D415_{3m}$	-3,213415	8,451179	89,201436	0,174850	0,0000056
$D415_{3,5m}$	14,881346	12,937068	89,600272	0,101297	0,0000025
$L515_{0,5m}$	-90,253647	11,729698	89,831525	0,014425	0,0000002
$L515_{0,75m}$	-56,201249	7,660510	89,812596	0,012077	0,0000001
$L515_{1m}$	-108,583925	5,192111	89,65577	0,018228	0,0000002
$L515_{1,25m}$	90,292230	8,332568	89,877903	0,004780	0,000000
$L515_{1,5m}$	-163,237338	2,228745	89,817268	0,031147	0,0000002
$L515_{1,75m}$	9,767191	3,500543	89,894598	0,018793	0,0000001
$L515_{2m}$	-141,670796	22,137824	89,943614	0,023941	0,0000001
$L515_{2,5m}$	-70,929531	20,429857	89,943384	0,008640	0,0000001
$L515_{3m}$	-118,381564	3,393077	89,746706	0,014246	0,0000001
$L515_{3,5m}$	-75,878716	3,218400	89,600172	0,006223	0,0000001
$SR305_{0,3m}$	-59,739436	29,423280	89,536397	0,149699	0,0000107
$SR305_{0,4m}$	-10,858720	15,751783	89,552815	0,295865	0,0000137
$SR305_{0,5m}$	21,009037	10,523013	88,977493	0,422890	0,0000305
$SR305_{0,75m}$	-178,152899	50,902432	88,997637	0,698460	0,0000902
$SR305_{1m}$	153,823401	22,655263	88,943926	0,317086	0,0000265
$SR305_{1,25m}$	-173,240353	33,826028	88,415494	0,558958	0,0000737
$SR305_{1,5m}$	-163,384550	47,076606	88,556564	1,007174	0,0002070