

The Depth Map Construction from a 3D Point Cloud

Pavel Chmelar^a, Ladislav Beran and Lubos Rejtek

Department of Electrical Engineering, Faculty of Electrical Engineering and Informatics, University of Pardubice, Czech Republic

Abstract. A depth map transforms 3D points into a 2D image and gives a different view of an observed scene. This paper deals with a depth map construction. It describes the whole process, how to transform any 3D point cloud into a 2D depth map. The described method uses 3D rotation matrixes and the line equation. This process allows to create the desired view from arbitrary point and rotation in an exploration space. Using of a depth map allows to apply image processing methods on depth data to get additional information about an ambient space.

1 Introduction

Depth maps (RGB-D) in mapping systems and robot navigation [1]-[4] are nowadays often used, but also in medical area for the respiratory motion detection [5], and in other areas for the motion and object detection.

For the 3D range scanning we are using a mobile platform with a laser and a camera, described in [6]. A laser spot is dispersed to a vertical line which allows measurement in the height range. In our previous work we introduced the new modelling of a colored laser line by using the laser Gaussian Mixture Model (GMM) [7]. The last paper describes an elimination of the barrel distortion correction [8]. These works contribute to the increase of the measurement points amount and measurement precision. The described 3D range scanning system is a part of the bigger project called ARES (Autonomous Research Exploration System) [9-11]. Our system is able to create a 3D point cloud, but up to now was not possible to create the depth map, which is useful for mapping systems mentioned above.

This paper describes the new algorithm for the depth map construction from a point cloud. This will give more information about depth data and it will allow to apply image processing methods 2D depth data.

2 Optical range finder

Our 3D range scanning system for the autonomous mapping, see Figure 1, is based on the triangulation method. The principle of the distance determination and the 3D point cloud constructions is introduced in [6].

The whole measuring device is pivotally carried by tripod. For the image capturing a high-quality Basler color camera with a resolution of 2590x1942 pixels is used. The used laser diode has an output power 200 mW, especially for good recognition after vertical swapping. The laser spot is dispersed by an optical filter with an

angle of 90 degrees. The 360 degrees rotation of the measuring head provides a powerful and fast stepping motor. The maximal measured distance is 15 m.



Figure 1. 3D range scanning system.

2.1 3D point cloud

The Figure 2 gives measurement results of one frame, which includes 80 measurement points.

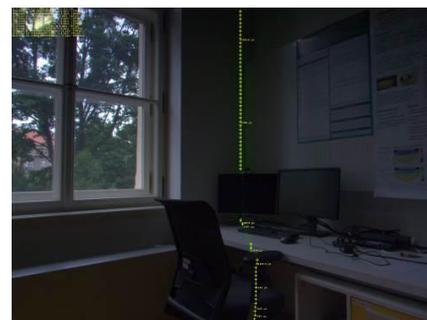


Figure 2. One measurement frame example.

The green laser line is segmented by the GMM, described in our previous work [7].

After one 360 degrees scan a 3D point cloud is created. An example of the 3D point cloud is in Figure 3.

^a Corresponding author: pavel.chmelar@student.upce.cz

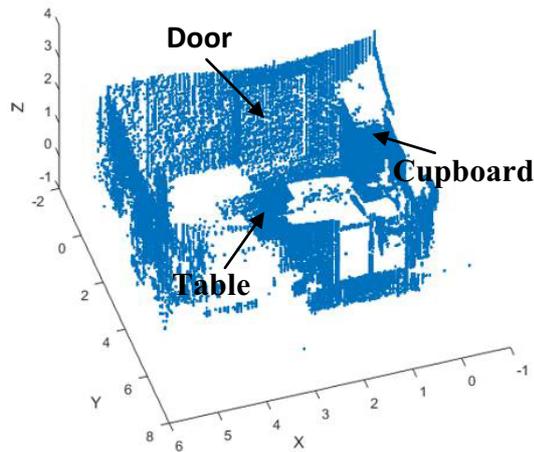


Figure 3. 3D point cloud.

This measured point cloud is used for the description of the depth map construction process in the following Section 3.

3 Depth map construction

For a depth map construction there are several initial parameters. The first is the camera position

$$P_C = (P_X, P_Y, P_Z) \quad (1)$$

The vector P_C (1) is a position from the depth map will be created. The second parameter is the camera field of view: horizontal H_{FOV} ; vertical V_{FOV} ,

$$C_{FOV} = \left(\frac{H_{FOV}}{2}, \frac{H_{FOV}}{2}, \frac{V_{FOV}}{2}, \frac{V_{FOV}}{2} \right) \quad (2)$$

The expression (2) is the field of view in the top, bottom, left and right area. It gives the used camera view range, which is able to cover. The last important parameter is the camera orientation defined by the following vector

$$C_O = (C_\alpha, C_\beta, C_\gamma) [^\circ] \quad (3)$$

This allows “to look” in the space in any desired orientation. The main advantage of the described algorithm is the possibility to set these parameters according our requirements, to create the desired resultant depth map.

For next described steps the two mathematical tolls are needed. For points rotations and transformations, the rotation matrixes in the 3D space (4), (5) and (6) are needed, described in paper [9]. To express the view range, the line equation (7) is used.

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (4)$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (5)$$

$$R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Symbols α , β and γ are rotation angles in corresponding axes X , Y and Z .

$$y = \tan(\alpha) \cdot x + b \quad (7)$$

The line parameter offset b is zero, $\tan(\alpha)$ is the line angle and x express the point where y is calculated.

3.1 3D point cloud initial preparations

In the initial stage the 3D point cloud is rotated according the camera orientation C_O (3). Although the input 3D point cloud is rotated, this rotation simplifies data selection process which describes the next Section 3.3. For the algorithm illustration $C_O = (0, 0, 90)$ is used. In Figure 4. are points rotated in C_O angles.

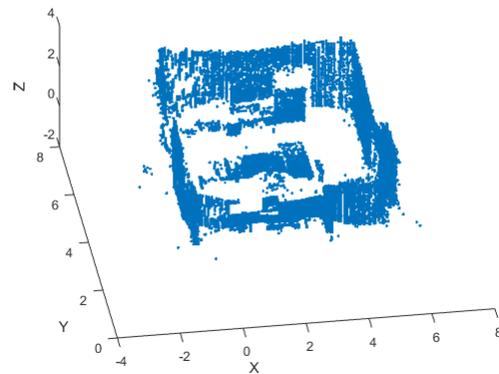


Figure 4. Rotated point cloud in angle of 90 degrees (Z axis)

3.2 Range view construction

The first algorithm step is the view range construction by using the line equation (7) and rotation matrixes in Y (5) and Z (6) axis. The line point step q_L is 1 cm. Firstly, the “V” (consist from two lines) with the V_{FOV} angle is created at the level $Z = 0$, subsequently the “V” is rotated in the angle of 90 degrees around Y axis (5). This rotated “V” is further rotated in angle of $H_{FOV}/2$ around the Z axis (6) in both sides (left and right). The whole range view is shifted to the camera position $P_C = (1.79, 3.3, 1.2)$. The result is shown in Figure 5. There are also orientation angles C_O (3) marked.

By using of the constructed view range, see Figure 6, the points for the depth map creation can be selected.

3.3 Range view data segmentation

To get the point of data selection describes Figure 7. There are marked four grey rectangles, which shows stages of the segmentation process.

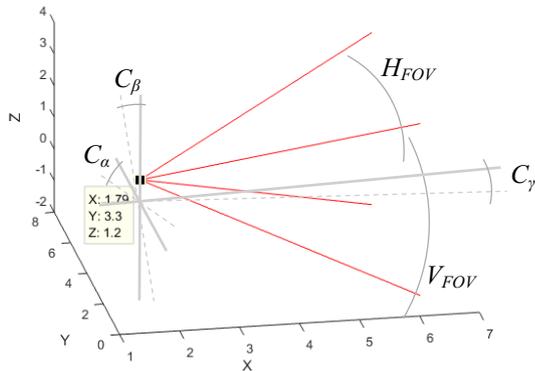


Figure 5. Constructed range view with rotation angles.

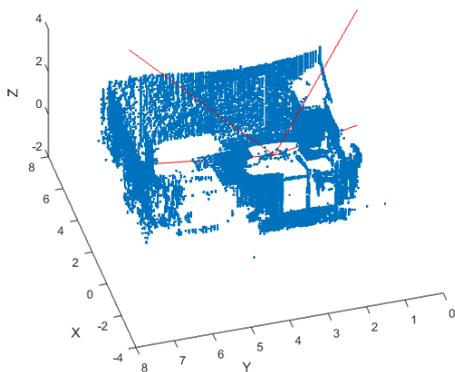


Figure 6. Constructed range view.

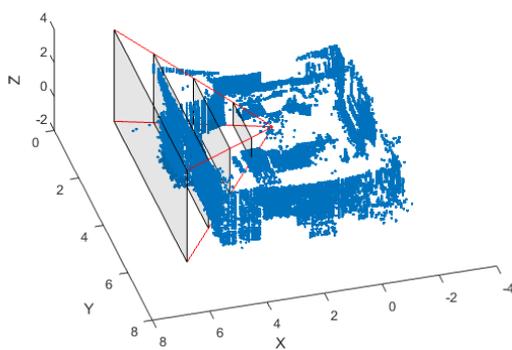


Figure 7. Points segmentation illustration.

The equation (8) describes the points which lies in grey rectangle R_{SYZ} in Y and Z axis

$$R_{SYZ}(n) = (PC(y) > R_S(n, y_{min})) \& (PC(y) < R_S(n, y_{max})) \& (PC(z) > R_S(n, z_{min})) \& (PC(z) < R_S(n, z_{max})), \quad (8)$$

where n is the view range line point from total N_L points (grey rectangle), PC is the point cloud. Values y and z means points in the appropriate axis Y and Z. Parameter R_S is segmentation rectangle and indexes min and max means the min position value in the appropriate axis and

vice versa. The final segmentation equation (9) by using equation (8) is the following

$$\bigcup_{n=1}^{N_L} PC(x) > R_S(n, x) \& R_{SYZ}(n). \quad (9)$$

The last unknown value is x which are points in the axis X. The whole segmentation process is logical sum of all points which comply the condition (9) (all points inside grey rectangles). The next Figure 8. shows the segmentation result with the marked view range.

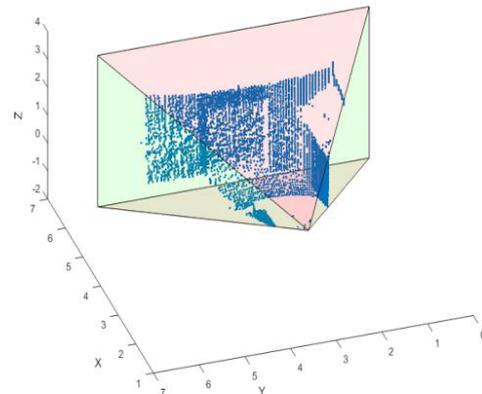


Figure 8. Segmented points from input point cloud.

3.4 Points depth image transformation

The final step is to transform points into the depth plane. The following Figure 9. illustrates points transformation. There are three red transformation lines for the demonstration.

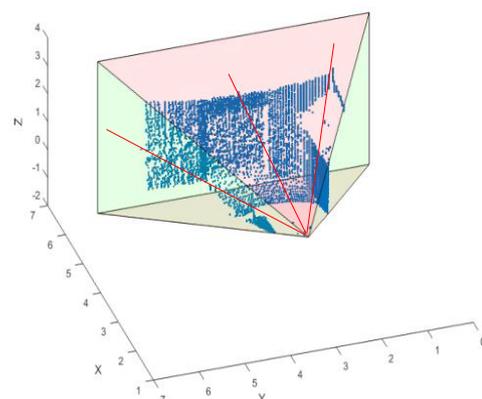


Figure 9. Illustration of point transformation.

Each point form the segmented point cloud has to be transformed into the final depth plane. The detailed transformation line description is in Figure 10, where the important transformation parameters are marked.

The distance parameters in Figure 10 marked by the uppercase letter means the direct distance to the depth plane and the parameters with the lower case letter are differences between two points. The parameters are the following: $D(x, y)$ is the transformation point P_t position in the depth plane; D_{PD} is the direct distance to depth

plane from the camera position \mathbf{P}_C ; d_y and d_z is the distance offset from the depth plane centre; angles φ_y and φ_z expresses the transformation line orientation; D_P , d_{Pz} , d_{PH} are parameters of the point triangle; D_{PD} , d_z , D_{PDH} constitute the second triangle (to the depth plane) and W and H is the depth map resolution.

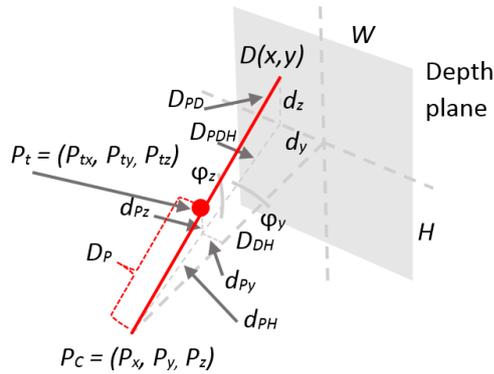


Figure 10. Point transformation.

The depth plane size is the last grey rectangle in Figure 7. The depth map resolution is possible to select by parameter q_{DM} , which express the depth map quantization. For the testing example parameter q_{DM} is equal to 1 cm, and then the final depth map horizontal resolution W is 724 px and the vertical resolution H is 662 px for the used point cloud.

The direct distance to the depth plane D_{DH} is known from the data range in X axis. Parameters d_{Pz} and d_{Pz} are the subtracted vectors \mathbf{P}_t and \mathbf{P}_C . The distance D_P to the measured point \mathbf{P}_t is the point norm of vector in \mathbf{R}^3

$$D_P = \sqrt{\sum |P_t - P_c|} \quad (10)$$

Angles φ_y and φ_z are calculated by using of goniometric functions. From known parameters is estimated the red line angle and calculated the distance D_{PDH} . This allows to express the red line point $D(x,y)$ and get exact position x and y in the depth plane. This process is done for all points in point cloud. If two points are projected on the same position the closer point is used.

The depth map pixel intensity is equal to point distance D_P quantized by 16 bit. The maximal intensity 2^{16} is equal to the maximal rangefinder range 15 m.

It is possible to assign all points directly in the depth plane from their position in space, but the main goal is to create the depth map from the camera view point \mathbf{P}_C . For this purpose, it is necessary to use the line transformation, showed in Figure 10. The point cloud typically contains points which can be covered by other objects like tables and etc. and these points is necessary to suppress in the resulting depth map. In this depth map construction process the Z-buffer method it is used [12]. The Z-buffer is known from computer graphics. When an object is rendered the Z-buffer ensure that the hidden points are not shown in the output frame. Practically the Z-buffer is the depth map. The point cloud does not give information about surfaces or points connection. The resulting depth

map is therefore checked for the farther points in all points neighbourhood. If in the neighbourhood area of 10 px ($q_{DM} = 1 \sim 10$ cm) lies a pixel with the distance farther than 0.5 m, these pixels are erased. These are estimated like covered by the closer pixel.

3.5 Resulting depth map

If the parameter q_{DM} is small like in this example 1 cm, it is possible to use the morphological operation “dilate” to increase the depth map point size. In the resulting depth map Figure 11. the dilatation by the element size 10 is used.

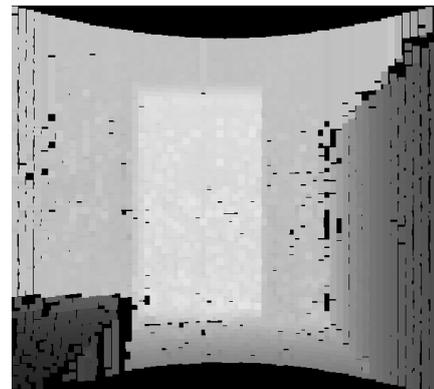


Figure 11. Resulting depth map.

The black pixels in top and bottom part of the resulting depth map are caused by the used input point cloud. There are missing points to display them in the created depth map. The darker pixels are closer to the camera position \mathbf{P}_C . The fineness and compact image is in farther distances where is the higher density of measurement points.

4 Results evaluation

The resulting depth map it is possible display also in surf graph, see Figure 12, which allows to “look” on the depth map in different view. The table, door and cupboard is obvious in this representation.

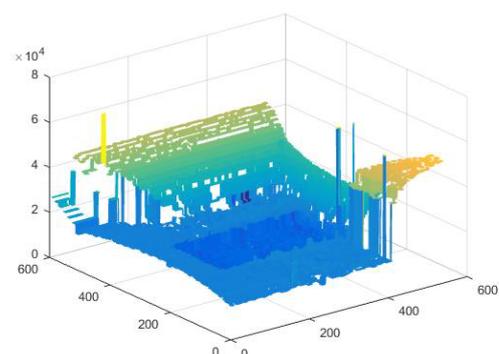


Figure 12. Surf depth map interpretation.

The next figures (Figure 13. 14. and 15.) show the depth map with the different camera position and

orientation angles $C_0 = (0,20,50)$. These figures shows that it is possible to use the different rotation and the position. The result quality is similar like the introduced example.

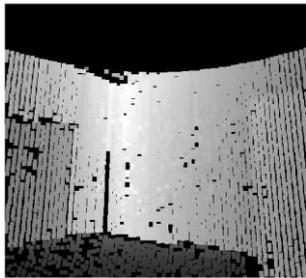


Figure 13. Depth Map in different orientation.

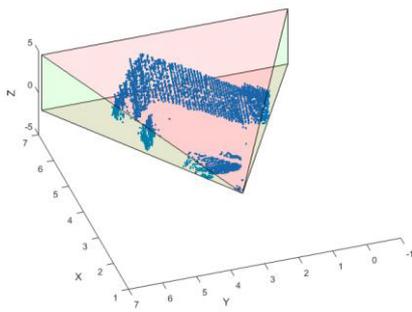


Figure 14. Segmented points for second depth map.

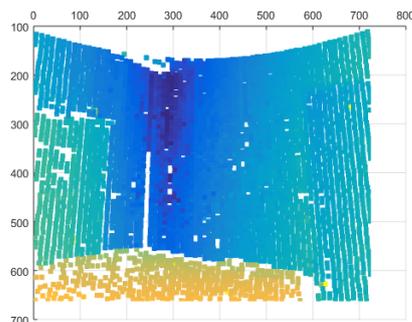


Figure 15. Surf interpretation for second depth map.

The Figure 15. shows the increasing distance to the edge of the room. The fineness effect is more obvious here. Points discontinuities in the closer distance are caused by the lower density of input point cloud. This is possible to solve by increasing the parameter q_{DM} , which has influence on depth map details.

5 Conclusion

In this paper we presented the algorithm for the depth map construction. We described the whole process with our optical rangefinder parameters. The main advantage of this algorithm lies in possibility to create a depth map from any point cloud with any camera position and any camera orientation.

There can be some black parts and holes in the resulting depth map. This disadvantage mainly depends

on the used point cloud and the selected detail parameter q_{DM} , which is necessary to choose with the deliberation.

Acknowledgement

The research was supported by the Internal Grant Agency of University of Pardubice, the project No. SGS_2016_022.

References

1. B.M.F. Silva, L.M.G. Gonçalves, A Fast Vis. Odom. and Mapp. Sys. for RGB-D Cam., *SBR LARS Robocontrol*, 55-60 (2014)
2. A. Aldoma, Z. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. Rusu, S. Gedikli, M. Vincze, Tutor.: Point cl. lib.: Three-dim. obj. recog. and 6 DOF pose est., *Robotics Automation Magazine IEEE*, **19**, 80-91 (2012)
3. H. Du, P. Henry, X. Ren, M. Cheng, D. Goldman, S. Seitz, D. Fox, Interact. 3D model. of ind. Enviro. with a consum. depth cam., *Proc. of the Int. Conf. on Ubiquitous Computing*, (2011)
4. K. Yousif, A. Bab-Hadiashar, R. Hoseinnezhad, 3D Reg. in Dark Environ. Using RGB-D Cam., *Digital Image Computing: Techniques and Applications (DICTA)*, 1-8 (2013)
5. C. Plagemann, V. Ganapathi, D. Koller, Real-tim. Identif. and loc. of bod. parts from depth imag., *IEEE International Conference on Robotics and Automation (ICRA)*, (2010)
6. P. Chmelar, M. Dobrovolny, The fus. of ultras. and optic. measurem. dev. for autonom. mapp., *Radioelektronika 23rd Int. Conf.*, pp.292-296, 2013
7. P. Chmelar, L. Beran, N. Kudriavtseva, The Las. Col. Det. for 3D Ran. Scan. Us. GMM, *Radioelektronika 25th Int. Conf.*, 248-253 (2015)
8. Chmelar P., Beran L., Rejfk L. N. Kudriavtseva, An Eff. Lens Distort. Correct. for 3D Range Scann. Syst., *In 57th Int. Symp. ELMAR*, 37-40 (2015)
9. L. Beran, P. Chmelar, M. Dobrovolny, Navig. of robot. platf. with us. inert. measure. unit and Dir. Cos. M, *ELMAR 56th Int. Symp.* 87-90 (2014)
10. L. Beran, P. Chmelar, and L. Rejfk, "Navigation of Robotics Platform using Monocular Visual Odometry," *Radioelektronika*, 2015 25th International Conference
11. L. Beran, P. Chmelar, L. Rejfk, Navig. of Robot. Platf. Us. Adv. Img. Proc. Navig. Meth., *VIPIMAGE*, (2015)
12. T. Theoharis, G. Papaioannou, E.-A. Karabassi, The Mag. of the Z-Buf.: A Surv., *Proc. of 9th International Conference on Computer Graphics, Visualization and Computer Vision WSCG*, (2001)