Plane Segmentation in Discretized Range Images

Viktor Kovacs[1], Gabor Tevesz[2]
Department of Automation and Applied Informatics
Budapest University of Technology
Budapest, Hungary
Email: {kovacsv[1], tevesz[2] }@aut.bme.hu

***Abstract*—In this paper we show a method for segmenting planes in discretized range images. Heavy quantization results in range images having a small number of different depth values. The range value encoded in each image pixel is not treated to be having random additive noise. For noise reduction purposes and feature extraction we use morphologic operations on each layer of the range image. Based on the features (layer skeletons) extracted from the images we show an algorithm for segmenting planes. We construct sets of linear skeleton segments and using a region growing technique we combine the linear segments into planar regions. We present segmentation results of simulated and captured real range images.**

I. Introduction

Advancements in 3D sensors and algorithms go hand in hand, supporting each other. In recent years 3D sensing products (time of flight, triangulation based sensors) have become more affordable thus finding their way to more applications. Based on the features of the sensor (horizontal, vertical, depth resolution, sampling rate, measurement errors, output data format) different post-processing techniques may be used to improve or filter data for specific applications.

Output data format may be ordered or unordered. The latter (point cloud) treats significance of each data point equal and considers only the coordinates of the points not their order. Range images are ordered structures, similar to conventional color images however pixels encode distance or depth coordinates. Color and range information may also be combined (RGB-D images). Layered depth images (LDI) describe not only the visible (from a given viewpoint) but the occluded surfaces as well. LDIs are mostly used in computer graphics to improve rendering performance by precalculating datasets.

Range images are based on regularly sampled data and implicitly contain information about adjacency thus surfaces, while point clouds need to be processed to reconstruct surfaces. On the other hand point cloud data implicitly comprise of world Cartesian coordinates (x,y,z), range image pixels must be projected back to (x,y,z) from image coordinates using the field of view (and may also be corrected for aberrations) and (u,v,z) data.

Segmentation is an important step in image understanding, incorporating the localization of parts of an image that fits a specific model or parts that fit different models. In order to map for example an indoors environment that consists of mostly planar surfaces, it is essential to detect planes in the sensor data. Detected planes may be used to interpolate missing data points, match viewpoints etc.

In this paper we deal with range images that comprise only a small number of different depth values due to quantization, compression, short stereo baseline or the sensor's principal of operation. We present an algorithm for extracting features from these range images such as planes. Our novel approach first extracts each layer of the input image to form binary images. One pixel wide skeleton lines are extracted from each layer and a skeleton segment graph is built up. We estimate the local surface normal vectors for the skeleton pixels. We further segment the skeletons to linear sections based on the estimated normal vectors. Linear segments are then connected by a region growing algorithm to localize planar surfaces. Other methods such as random sampling or Hough transformation based techniques are misled by quantization error.

The organization of the paper is as follows. In section II we present related work. Section III. deals with skeleton extraction, local surface normal estimation and plane extraction. In section IV we show results of processing simulated data and captured images. Section V contains conclusions.

II. Related work

3D sensing (time of flight cameras, structured light based reconstruction) is extensively used in 3D scene (urban or indoors) reconstruction, object detection, autonomous vehicle navigation or other robotics applications such as SLAM (Simultaneous Localization and Mapping). In case of artificial environments a large portion of surfaces can be described as planes thus plane detection is widely used as a first step in range image understanding.

Hough transformation is widely used for pattern recognition in image processing, especially for line or circle detection. Based on the n number of parameters, an n dimensional discretized accumulator space is formed. Detected features in the original image vote for all possible parameter sets that describe the feature. In case of lines $y = ax + b$ each edge or active point vote for all the a, b parameters that contain the (x, y) point. Finally the accumulator space is evaluated for local maximums, parameter sets that were voted by most features. It is clear that the number of parameters and the parameter space resolution restricts the applicability, having more than two or three dimensional accumulator spaces leads to huge memory requirements. In [1] Iocchi et al. present 3D Hough transformation used for plane fitting. Every point corresponds to a curve in the accumulator space. Still in case of carefully chosen discretization of parameters it is possible to achieve real-time performance. Borrmann et al. [2] proposed a new accumulator space design (accumulator ball) in order to improve performance by regularizing parameter resolution.

Weingarten et al. [3] used laser scanner to acquire unordered datasets of indoors environments. Data were decomposed to cubic regions and stored in a dynamic list structure. The side length of the cubes were about 30 cm, each containing between 1 and 51 vertices. Random Sample Consensus (RANSAC [4]) was used to fit planes in each cubic region. Model parameters were evaluated based on the least squares method. The advantage over the PCA method is calculation speed. For the pixels, supporting the plane model, a 2D quad bounding box is calculated. Neighboring quads are merged together based on matching orientation and translation. In [5] authors used this approach in a SLAM application. The plane segments were used as landmarks for a SLAM. In case of heavily quantized range images RANSAC methods are not easily applicable at first. As each layer is described perfectly by a plane perpendicular to the camera axis, sampling must be consciously designed. The region growing segmentation step in our method could be substituted by RANSAC, by taking samples from skeleton pixels.

A fast plane detection algorithm was presented by Poppinga et al. in [6]. This method is based on region growing and local plane fitting as an eigenvector problem, however in order to improve efficiency, instead of a naive implementation, an incremental calculation method was proposed. The algorithm also exploits the neighborhood information that is available in range images. The authors also present a polygonalization method that is applied in 2D image space and later transferred to a 3D model.

A coarse to fine method for plane segmentation is used in [7]. First surface elements (surfels) are extracted at multiple resolutions from coarse to fine. In each step surfels are associated to planar regions that were detected in the previous step. Planes are created by grouping new unassociated surfaces to planes by Hough-transform. Connected planes are refined by RANSAC to find the best fitting plane and reject outliers.

Hulik et al. [8] modified three known depth image segmentation methods and proposed two new algorithms. They compared all five methods for accuracy and time consumption. They showed that while accuracy measures gave similar results, visual comparison showed significant differences.

Guillaume et al. [9] presented a surface segmentation method based on curvature tensors. Curvature is a widely used local descriptor for surfaces due to its invariant nature. Authors segmented triangle meshes into constant curvature regions. Based on curvature not only planar (infinite curvature) but other special surface types such as spherical or cylindrical surfaces may be easily segmented. Again, the estimation of curvature in discretized depth maps is a challenge.

In a previous work [10] we proposed a method to smooth range images that suffer from a significant amount of quantization noise. After smoothing we used a Hough-space method, orientation histogram to cluster local normals. Segmenting in Hough-space gave us parallel plane segments. We segmented point distribution along the normal axis to identify parallel planes.

We presented a method to identify salient points and corners using our layer based method in [11]. We used higher level heuristics to identify points that meet certain set of criteria in order to recognize corners and classify their type.

III. Plane segmentation

In order to detect planes we first create iso-depth skeletons from the layers. Next we approximate the local surface normals, then segment the skeletons to linear sections. Finally we grow planar regions from the linear segments.

A. Skeleton extraction

Our method extracts each layer (given by the available range values) from the image and processes these layers individually. These layers are considered as binary images. In order to reduce noise we remove small connected regions and apply morphological dilation to the images. The connected components algorithm is used to identify small regions. The region size threshold and the amount of dilation must be selected based on the noise. After dilation skeletons are extracted [12]. The applied algorithm has excellent properties for our purposes such as few side branches, compared to other simple thinning algorithms.

Layer skeletons are one pixel wide lines resembling the features of each layer. These skeletons are similar to what we would see in case of extracting edges from an ideal but quantized range image, however the skeletons due to the dilation-erosion step, are more noise tolerant. Also due to the dilation as an unfortunate side effect, sharp corners become curved.

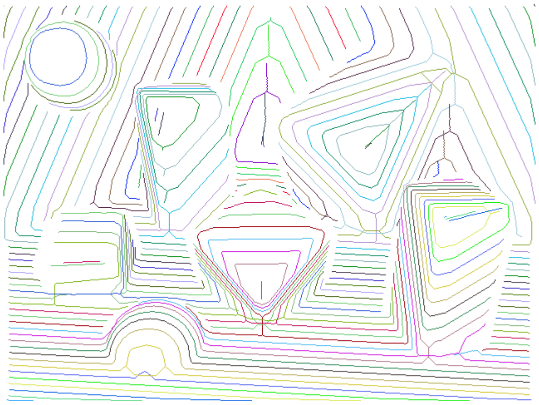

Fig. 1. Skeleton map generated from a quantized range image (44 layers)

Skeletons are broken into segments, which connect junctions or endpoints and do not contain other junctions. Short segments are likely to be unwanted side-branches. Segments containing pixels less than n_{MinSeg} are removed and segments are reevaluated, reconnected.

B. Local surface normal estimation

In case of ideal range images it is possible to sample a small region of pixels in image space, back-project them to world coordinates. The local surface normals may be estimated by PCA, the least significant direction points to the normal direction. Based on the error of the fitting it may be decided weather the sample is part of a plane.

Plane detection in range images having few depth values (quantized) is more challenging. In such case most of the estimated normals would be directed towards the camera $(0, 0, -1)$

direction as the layers are planar, having the same depth value. Similarly RANSAC based methods would find planar patches in the x-y plane to be best fitting. Noise model should not be considered random, quantization error is systematic. Pixels should not be handled equally: those that are on the edge or the center of quantization layers should be used for sampling. One option is to interpolate the pixel values within a layer [10] and estimate the local normals. Still, the noise might be high and the interpolation may be costly and might introduce other artifacts.

A 2D tangential orientation α is calculated for each pixel along all the segments by fitting a line to given number of $n_{FitLine}$ neighbors of the skeleton pixel. The tangent vector v is given by the eigenvector of the covariance matrix corresponding to the largest eigenvalue. This 2D orientation α equals the 3D tangential projected to the x-y plane. The b binormal direction is needed to estimate the local surface normal n.

$$C = cov(X,Y) = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \tag{1}$$

$$\lambda_{1,2} = \frac{\sigma_x^2 + \sigma_y^2 \pm \sqrt{(\sigma_x^2 + \sigma_y^2)^2 - 4(\sigma_x^2\sigma_y^2 - \sigma_{xy})}}{2} \tag{2}$$

$$\lambda^* = \max(|\lambda_1|, |\lambda_2|) \tag{3}$$

$$\mathbf{v}^* = \begin{bmatrix} -(\sigma_x^2 - \lambda^*)/\sigma_{xy} \\ 1 \end{bmatrix} \tag{4}$$

$$\mathbf{v} = \frac{\mathbf{v}^*}{|\mathbf{v}^*|} \tag{5}$$

$$\alpha = \arctan(v_2/v_1) \tag{6}$$

In order to determine b direction let us consider a P plane that is perpendicular to the previously defined tangent vector v. We look for the b binormal in P. We evaluate the signed distance d eq. (7) from P (given by the selected point p_0 and the tangent vector v) to all the p skeleton points along the segments in the next layer. We select a point where the signed distance changes sign and has the shortest Euclidean distance $d(p_0, p)$. This p^* point is considered adjacent to p_0. The binormal is given by $b = p^* - p_0$. The normal n is given by $n = b \times v$. Normals are flipped if needed to point to the front faces as the direction of v is undetermined. All vectors are normalized to unit vectors. To improve noise tolerance, a combination of several binormals may also be used. Not only the following but several adjacent layers' binormals may be used to estimate n. Note that in this case details such as edges, corners may be filtered.

$$d(p_0, p) = v \cdot (p - p_0) \tag{7}$$

where v is the tangent at point p_0.

The algorithm deals with pixels of layer skeletons, however to help visualization we assigned each pixel in the image space to the closest skeleton pixel in the same layer.

C. Plane extraction

In our model we assume that quantization levels are given by $x - y$ planes, perpendicular to the z axis. Considering P_1 $(\mathbf{n_1}, \mathbf{p_1})$ and P_2 $(\mathbf{n_2}, \mathbf{p_2})$ planes. P_1 is arbitrary, P_2 represents the quantization layer. Their intersection line is $\mathbf{l}$, given by:

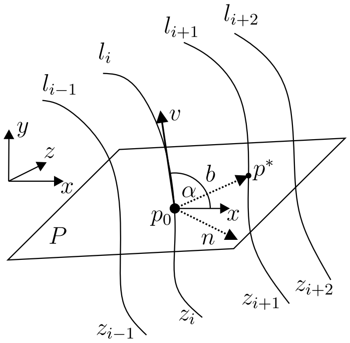

Fig. 2. l_i: layer skeletons, v tangent, b binormal, n normal direction

$$\mathbf{l} = \mathbf{p_0} + t\mathbf{v} = (x, y, z)^T \tag{8}$$

$$\mathbf{v} = \mathbf{n_1} \times \mathbf{n_2} \tag{9}$$

$$\mathbf{n_1} \cdot \mathbf{p_0} = d_1 = \mathbf{n_1} \cdot \mathbf{p_1} \tag{10}$$

$$\mathbf{n_2} \cdot \mathbf{p_0} = d_2 = \mathbf{n_2} \cdot \mathbf{p_2} \tag{11}$$

Assuming $\mathbf{n_2} = (0, 0, z)^T$, P_1, P_2, l all contain $\mathbf{p_0}$, leads to:

$$y = ax + b = -\frac{n_x}{n_y}x\frac{d_1 - n_z z}{n_y} \tag{12}$$

Taking perspective projection into account in the image space (u, v) where $u = x/z$, $v = y/z$:

$$v = -\frac{n_x}{n_y}u\frac{d_1 - n_z z}{n_y} \tag{13}$$

Meaning that by changing z quantization edge position, it does not change the orientation of the line in image space, that is representing P_1 plane. Thus all $P(\mathbf{n}, \mathbf{p})$ planes are represented by n_x/n_y slope lines.

As a next step in the proposed algorithm we extract linear sections of the skeleton segments. To do this we first define $\beta(p)$ breakage angle for each skeleton pixel by fitting a line to n_β number of pixels to both the left and right neighbors of the pixel. The intersection angle is stored for each skeleton point. Next we apply a region growing algorithm for each segment: select the skeleton pixel where (14) the sum of all i adjacent pixels' breakage angle is minimal. In case this sum is larger than a β_{min} threshold, the segment is omitted. A linear segment is formed by adding p_0 and its adjacent skeleton point until the angle between the normal of the segment and the normal of the examined point is less than a given γ_{max} threshold. Each time a point is added, a given weight is associated to modify the normal direction and center point for the linear segments. The weight is given by the 2D Euclidean distance of the point and the center of the segment. $w(d_{2D}) = (\sqrt{(2\pi)}\sigma)^{-1}\exp(-d_{2D}/2\sigma^2)$ where σ is a predefined constant.

$$p_0 = \underset{p \in S}{\operatorname{argmin}} \sum_{q \in N_i(q)} \beta(q) \tag{14}$$

The next step is to grow parallel linear segments into planes. First the longest unprocessed linear segment l_0 is

LinearSegment 2 Linear segment extraction algorithm

```
input: $S$
output: $P$
while $S \neq \emptyset$ do
    $lookl, lookr \leftarrow$ true
    $n \leftarrow 1$
    $p_0 \leftarrow$GetFirstPoint($S$)
    $L_i \leftarrow p_0$
    while not $finished$ do
        $l \leftarrow$GetNthLeftNeighbor($S$,$p_0$,$n$)
        $r \leftarrow$GetNthRightNeighbor($S$,$p_0$,$n$)
        if $lookl$ and Angle($l.n$, $L.n$) $< \gamma_{max}$ then
            $L_i \leftarrow l$, Weight($L_i.p, l$)
        else
            $lookl \leftarrow$ false
        end if
        if $lookr$ and Angle($r.n$, $L.n$) $< \gamma_{max}$ then
            $L_i \leftarrow r$, Weight($L_i.p, r$)
        else
            $lookr \leftarrow$ false
        end if
        Update $L_i.n$, $L_i.p$
        $finished \leftarrow$ (not ($lookl$ or $lookr$))or( $\{l, r\} = \emptyset$)
        $n \leftarrow n + 1$
    end while
    if Count($L_i > n_{min}$) then
        $L \leftarrow \{L, L_i\}$
    end if
end while
```

selected. Next the linear segments on the neighboring layers $L_{l_0 \pm j}$ are searched for similar normal directions and distances. Linear segments that fulfill the requirements are added to the plane (P_i). Each time a linear segment is added a weight is also given to modify the plane parameters. $w = 1/\sqrt{2\pi}\sigma \exp(-\text{length}_{3D}(l_i)^2/2\sigma^2)$, $\sigma = \text{length}_{3D}(l_0)$. Next the adjacent layers are evaluated and so on. The process stops when no more fitting segments are found in the last c_{max} layers. A new (P_{i+1}) plane segmentation starts.

$$\text{length}_{3D}(l) = \max_{p,q \in l} d(p, q) \tag{15}$$

IV. Results

In this section we investigate the accuracy of the algorithm for plane segmentation. For evaluation purposes first we use simulation images at different discretization levels with ground truth data. The algorithm was also tested on real captured data.

A. Simulation

Simulated scenes (Fig. 3.) were constructed with ground truth data to test the algorithm at different quantization levels. The scenes contained several distorted cubic and spherical objects. In order to simulate quantization a non-linear, tangential function was used. Although the image did not suffer from any noise, we used the same parameters (dilation) as for real images. This resulted in significant smoothing of skeletons around edges (Fig. 1). Figure 4. shows estimated local surface normals for skeleton pixels. Due to the quantization of the layer skeletons' pixel positions, especially near 0° and 90° orientation, local surface normals tend to fluctuate. By increasing the number of pixels used for α estimation, this effect may be handled. Normal estimation on non-planar surfaces (spheres) showed good results. The resulting segmentation of the algorithm can be seen in Fig.5. Each color represents a segmented planar surface. Results show uncertainty near crease edges, in some cases these regions are oversegmented. By choosing an appropriate γ_{max} value it is possible to restrict segmentation from edges. The normals estimated for these individual planes are shown in Fig. 6 .

By increasing the depth resolution layers become less continuous causing linear skeleton segments to break into several pieces and distributed on several adjacent layers. This affects

PlaneSegment 4 Plane segmentation algorithm

```
input: $L$
output: $P$
loop
    $l_0 \leftarrow \underset{l \in L}{\text{argmaxlength}}(l)$
    if length($l_0$)¡$n_{min}$ then
        Exit.
    end if
    $P_i \leftarrow l_0$
    $j \leftarrow 0$
    $c \leftarrow 0$
    loop
        $L_s = \{l | Angle(P_i, l) < \gamma_{max} \wedge |d(P_i, l)| \leq d_{max}\}$
        $l \in L_{l_0+j} \cup L_{l_0-j}$
        if $\#L_s > 0$ then
            $P_i \leftarrow L_s$, Weight($P_i, L_s$)
            $c \leftarrow 0$
        else
            $c \leftarrow c + 1$
            if $c \geq c_{max}$ then
                End loop
            end if
        end if
    end loop
    if $\#P_i > p_{min}$ then
        $P \leftarrow P_i$
    end if
end loop
```

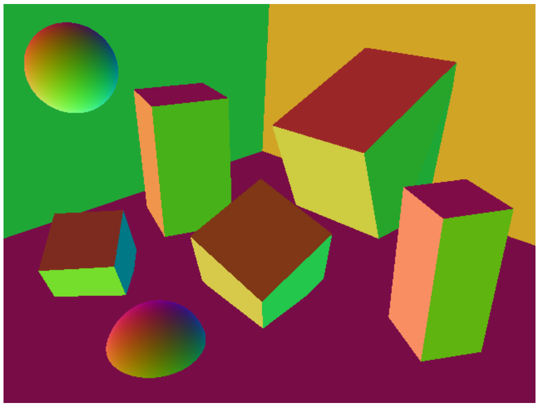

Fig. 3. Simulated scene, colors represent normal direction (RGB:XYZ)

the performance of the algorithm significantly as increased number of adjacent layers are needed to be searched for skeleton segments. In case of higher depth resolution range images, other conventional methods should be used.

In case of decreasing the number of layers the algorithm still gives acceptable results. The number of planar segments identified depends on the size of the regions and their orientation respect to the camera. A plane near perpendicular to the camera would be described by only a few layer segments thus increasing localization uncertainty.

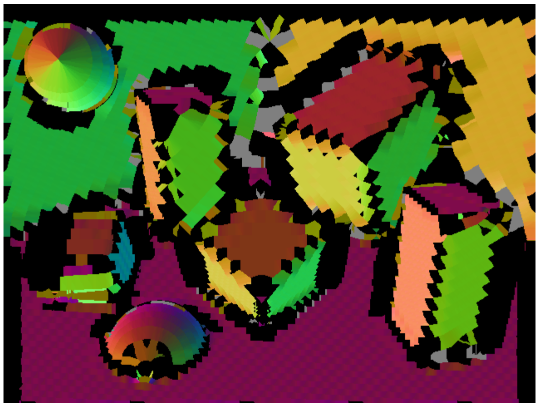

Fig. 4. Estimated local surface normals

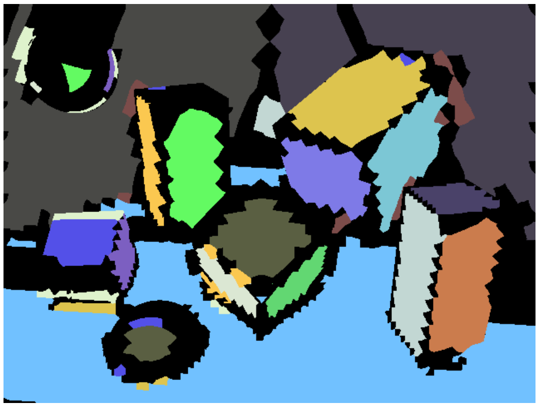

Fig. 5. Plane segmentation results

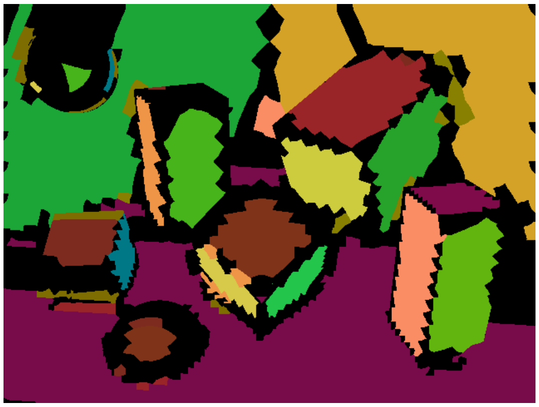

Fig. 6. Reconstructed, segmented normal direction image, 44 layers, steep regions were omitted

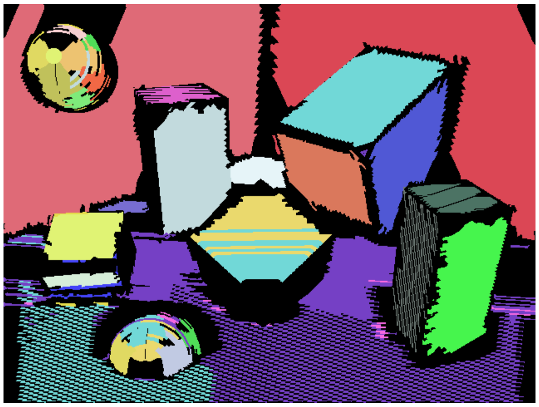

Fig. 7. Oversegmentation at 192 layers

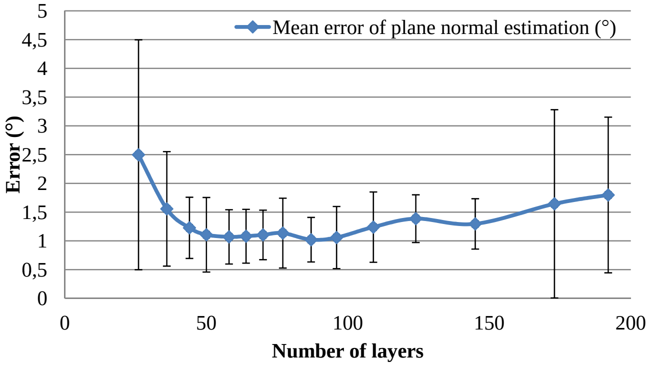

Fig. 8. Normal approximation error at different depth resolution

Figure 8. shows the mean error of the normal vector estimation for planes in the sample image. The measurements are based on different locations that are meant to be part of planar surfaces. In case the selected point was not identified as part of a plane, the value was omitted. As the number of layers increase the mean error drops, until about 80-100 layers. Having even more layers slowly increases the error as layers become more fragmented, less continuous (Fig. 7.). The optimal number of layers depend on the image contents and resolution. In case of the mean plane errors we see lower estimation error, as low as 1°.

Figure 9. shows the distribution of normal error for individual pixels that are part of a segmented plane, and for different quantization levels. We see that for less range layers the error falls faster, giving more accurate estimations. For more layers we see less drop in the error distribution due to the problem of quantization in image space: layer skeletons are very close thus estimation is more uncertain. The most significant portion of the estimation error lies in the 0-5° region.

B. *Real data*

We also evaluated the algorithm by using real data acquired by the Microsoft Kinect sensor. Optical distortions were not corrected. We examined the algorithm by further reducing the number of depth layers. Due to noise, oversegmentation is more likely compared to simulated images. By selecting algorithm parameters for a specific image acquisition system and application, segmentation errors may be greatly reduced.

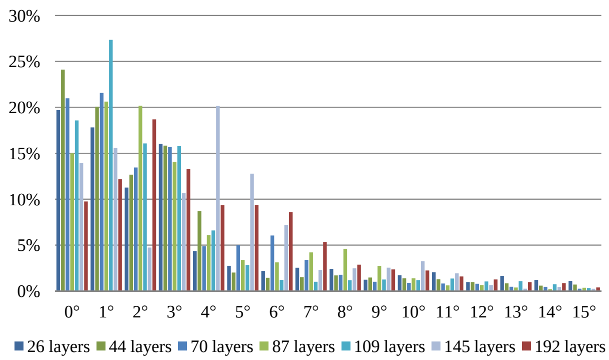

Fig. 9. Normal approximation error distribution on planes

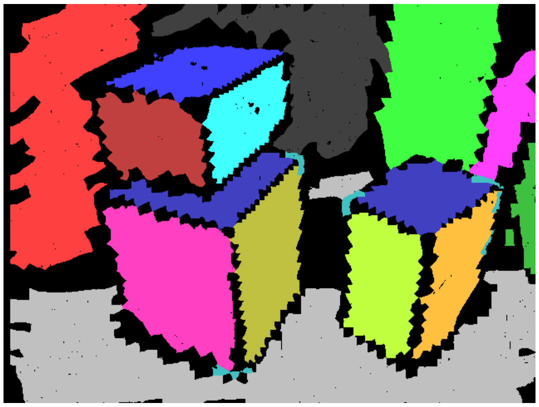

Fig. 10. Segmentation of a captured range image, consisting of only 41 layers

Figure 10. shows a segmentation result of a real image consisting of only 41 depth levels. All planar regions were identified, however small regions were omitted due to uncertainty. The top face of the two boxes in the image were nearly in an identical plane thus these were segmented as one. The algorithm may be easily modified to separate these regions. However the farthest part of the floor would also be segmented in a different plane.

V. Conclusion

In this paper we presented an algorithm capable of identifying planar surfaces in heavily quantized range images. Such data is not suitable for traditional algorithms as quantization error treated as random noise leads to undesired results. Our proposed algorithm utilizes a low amount of different depth values, creates skeleton structures from layers and estimates local surface normals. Finally skeletons are grown into planar regions.

We evaluated the performance of the algorithm by using simulated images as ground truth. We examined the accuracy at several quantization levels and concluded the algorithm works well between 40-100 layers - depending on the contents of the image. High and low slope planes may not be identified properly. High slope regions should be handled in the traditional way as these parts of the image are not considered heavily quantized.

Real captured images were also used in the evaluation process, where the algorithm showed excellent segmentation results even in images suffering from significant noise (uncertain transient between layers).

Acknowledgment

This work was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013) organized by VIKING Zrt. Balatonfured. This work was partially supported by the Hungarian Government, managed by the National Development Agency, and financed by the Research and Technology Innovation Fund (grant no.: KMR-12-1-2012-0441).

References

[1] L. Iocchi, K. Konolige, and M. Bajracharya, "Visually realistic mapping of a planar environment with stereo," in *Experimental Robotics VII*, ser. ISER '00. London, UK, UK: Springer-Verlag, 2001, pp. 521–532.

[2] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nchter, "The 3d hough transform for plane detection in point clouds: A review and a new accumulator design," *3D Research*, vol. 2, no. 2, 2011.

[3] J. Weingarten, G. Gruener, and R. Siegwart, "A Fast and Robust 3D Feature Extraction Algorithm for Structured Environment Reconstruction," in *None*, 2003.

[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[5] J. Weingarten and R. Siegwart, "3d slam using planar segments," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Oct 2006, pp. 3062–3067.

[6] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak, "Fast plane detection and polygonalization in noisy 3d range images," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept 2008, pp. 3378–3383.

[7] B. Oehler, J. Stueckler, J. Welle, D. Schulz, and S. Behnke, "Efficient multi-resolution plane segmentation of 3d point clouds," in *Intelligent Robotics and Applications*, ser. Lecture Notes in Computer Science, S. Jeschke, H. Liu, and D. Schilberg, Eds. Springer Berlin Heidelberg, 2011, vol. 7102, pp. 145–156.

[8] R. Hulik, V. Beran, M. Spanel, P. Krsek, and P. Smrz, "Fast and accurate plane segmentation in depth maps for indoor scenes," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct 2012, pp. 1665–1670.

[9] L. Guillaume, D. Florent, and B. Atilla, "Curvature tensor based triangle mesh segmentation with boundary rectification," in *Computer Graphics International, 2004. Proceedings*, June 2004, pp. 10–25.

[10] V. Kovacs, "Landmark detection in low depth resolution range and intensity images," in *Automation and Applied Computer Science Workshop (AACS 2012)*, June 2012, pp. 169–181.

[11] V. Kovacs and G. Tevesz, "Corner detection and classification of simple objects in low-depth resolution range images," *Periodica Polytechnica, EECS*, vol. 57, no. 1, pp. 9–17, aug. 2013.

[12] G. Németh and K. Palágyi, "2d parallel thinning algorithms based on isthmus-preservation," in *ISPA 2011: 7th International Symposium on Image and Signal Processing and Analysis: Dubrovnik, Croatia, 4 - 6 September 2011. IEEE*, 2011, pp. 585–590.