A Data Structure for the 3D Hough Transform for Plane Detection

Dorit Borrmann * **Jan Elseberg** * **Kai Lingemann** ** **Andreas Nüchter** *

* *Jacobs University Bremen, School of Engineering and Science, Campus Ring 1, 28759 Bremen, Germany*
e-mail: d.borrmann@jacobs-university.de
** *University of Osnabrück, Institute of Computer Science, Knowledge-Based Systems Research Group, Albrechtstr. 28, 49069 Osnabrück, Germany*

Abstract:
The Hough Transform is a well-known method for detection of parametrized objects. It is the de facto standard for detecting lines and circles in 2-dimensional data sets. For 3D it has attained little attention so far. Apart from computation costs, the main problem lies in the representation of the accumulator: Usual implementations favor geometrical objects with certain parameters due to uneven sampling of the parameter space. In this paper we present a novel approach to design the accumulator focusing on achieving the same size for each cell. The proposed accumulator is compared to previously known designs.

Keywords: Hough Transform, laser scans, plane detection, indoor mapping

1. INTRODUCTION

One of the main research topics in mobile robotics attends to creating maps of the robot's surroundings. Mapping is often achieved by matching separately collected partial views, for example laser scans. A laser scan is a set of distance values. Each laser scan, taken by a robot at different positions, represents a small part of the environment, the part that is observable from the current pose. The process of joining these partial views into one map is referred to as *scan matching*. Scan matching is subject to matching errors, sensor noise and systematic errors in the scans. The structure of indoor environments, however, usually comprises a large amount of planar surfaces. This knowledge can be used to help reconstruct the real structure of the environment.

Plane extraction, or plane fitting, is the problem of modeling a given 3D point cloud as a set of planes that ideally explain every data point. The RANSAC algorithm is a general, non-deterministic algorithm that iteratively finds an accurate model for observed data that may contain a large number of outliers (cf. Fischler and Bolles (1981)). Schnabel et al. (2007) have adapted RANSAC for plane extraction and found that the algorithm performs precise and fast plane extraction, but only if the parameters have been fine-tuned properly. Bauer and Polthier (2008) use the radon transform to detect planes. The idea and the speed of the algorithm are similar to that of the Hough Transform, however the accuracy of the detected planes is significantly worse. Other plane extraction algorithms are highly specialized for a specific application and are not in widespread use for miscellaneous reasons (cf. Borrmann and Elseberg (2009)).

2. THE 3D HOUGH TRANSFORM

The Hough Transform (Hough (1962)) is a method for the detection of parametrized objects. Typically used for lines and circles, we will focus on the detection of planes in 3D point clouds here. Planes are commonly represented by the signed distance ρ to the origin of the coordinate system and the slope m_x in direction of the x-axis and m_y in the direction of the y-axis, respectively:

$$z = m_x x + m_y y + \rho.$$

To avoid the arising problems due to infinite slopes when trying to represent vertical planes, another usual definition uses normal vectors. A plane is thereby given by a point $\mathbf{p}$ on the plane, the normal vector $\mathbf{n}$ that is perpendicular to the plane and the distance ρ to the origin

$$\rho = \mathbf{p} \cdot \mathbf{n} = p_x n_x + p_y n_y + p_z n_z = \rho.$$

Taking the angles between the normal vector and the coordinate system into consideration the coordinates of $\mathbf{n}$ can be factorized to

$$p_x \cdot \cos\theta \cdot \sin\varphi + p_y \cdot \sin\varphi \cdot \sin\theta + p_z \cdot \cos\varphi = \rho, \quad (1)$$

with θ the angle of the normal vector on the xy-plane and φ the angle between the xy-plane and the normal vector in z direction. φ, θ and ρ define the 3-dimensional Hough Space (θ, φ, ρ space) such that each point in the Hough Space corresponds to one *plane* in $\mathbb{R}^3$.

To find planes in a set of points we need to calculate the Hough Transform for each point. Given a point $\mathbf{p}$ in Cartesian coordinates, we have to find all the planes the point lies on, i. e., find all the θ, ϕ and ρ that satisfy Eq. (1). Marking these points in Hough Space leads to a sinusoid curve. The intersections of two curves in Hough space denote the planes that are rotated around the line built by the two points. Consequently, the intersection of three curves in Hough space corresponds to the polar coordinates defining the plane spanned by the three points.

Given a set P of points in Cartesian coordinates, we have to transform all points $\mathbf{p}_i \in P$ into Hough Space. The more curves intersect in $\mathbf{h}_j \in (\theta, \varphi, \rho)$, the more points lie on the plane represented by $\mathbf{h}_j$ and the higher is the probability that $\mathbf{h}_j$ can actually be extracted from P.

3. STANDARD HOUGH TRANSFORM

For practical applications Duda and Hart (1971) propose to discretize the Hough Space with ρ', φ' and θ' denoting the extend of each cell in the according direction in Hough Space. A data structure is needed to store all these discretized cells and a score parameter for every single cell. This data structure, called the accumulator, is described in more detail in Section 5. For each point p_i we accumulate all the cells that are touched by its Hough Transform. The incrementation process is often referred to as voting, i.e., each point votes for all sets of parameters (ρ, φ, θ) that define a plane on which it may lie. The cells with the highest values represent the most prominent planes, the plane that covers the most points of the point cloud.

Once all points have voted, the winning planes have to be determined. Due to the discretization of the Hough Space and the noise in the input data it is expedient to search not only for one cell with a maximal score but for the maximum sum in a small region of the accumulator. Kiryati et al. (1991) use the standard practice for peak detection. In the sliding window procedure a small 3-dimensional window is defined that is designed to cover the full peak spread. The most prominent plane corresponds to the center point of a cube in Hough Space with a maximum sum of accumulation values. The steps of the algorithm are outlined in Algorithm 1.

Algorithm 1 Standard Hough Transform (SHT)

```
for all points p_i in point set P do
    for all cells (ρ, φ, θ) in accumulator A do
        if point p_i lies on the plane defined by (ρ, φ, θ) then
            accumulate cell A(ρ, φ, θ)
        end if
    end for
end for
Search for the most prominent cells in the accumulator, that define the detected planes in P
```

4. RANDOMIZED HOUGH TRANSFORM

Due to its high computation time, the SHT is rather impractical, especially for real-time applications. Xu et al. (1990) describe the Randomized Hough Transform (RHT) that diminishes the number of cells touched by making use of the fact that a curve with n parameters is defined by n points.

Taking the example of planes, three points from the input space can be mapped onto one point in the Hough Space. This point is the one corresponding to the plane spanned by the three points. In each step of the procedure three points $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ are randomly picked from the point cloud. The plane spanned by the three points is calculated as $\rho = \mathbf{n} \cdot \mathbf{p}_1 = ((\mathbf{p}_3 - \mathbf{p}_2) \times (\mathbf{p}_1 - \mathbf{p}_2)) \cdot \mathbf{p}_1$. φ and θ are calculated as explained in Section 2 and the corresponding cell $A(\rho, \varphi, \theta)$ is accumulated. If the point cloud consists of a plane with ρ, φ, θ, after a certain number of iterations there will be a high score at $A(\rho, \varphi, \theta)$.

When a plane is represented by a large number of points, it is more likely that three points from this plane are randomly selected. Eventually the cells corresponding to actual planes receive more votes and are distinguishable from the other cells. If points are very far apart, they most likely do not belong to one plane. To take care of this and to diminish errors from sensor noise a distance criterion is introduced: $\mathrm{dist}_{\max}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) \leq \mathrm{dist}_{\max}$, i.e., the minimum point-to-point distance between $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ is below a fixed threshold; for maximum distance, analogous. The basic algorithm is structured as described in Algorithm 2.

Algorithm 2 Randomized Hough Transform (RHT)

```
while still enough points in point set P do
    Randomly pick three points p_1, p_2, p_3 from the set of points P
    if p_1, p_2 and p_3 fulfill the distance criterion then
        Calculate plane (ρ, φ, θ) spanned by p_1 ... p_3
        Accumulate A(ρ, φ, θ) in the accumulator space.
        if the cell |A(ρ, φ, θ)| equals threshold t then
            (ρ, φ, θ) parameterize the detected plane
            Delete all points close to (ρ, φ, θ) from P
            Reset the accumulator
        end if
    else
        continue
    end if
end while
```

The Randomized Hough Transform has several main advantages. Not all points have to be processed, and for those points considered no complete Hough Transform is necessary. Instead, the intersection of three Hough Transform curves is marked in the accumulator. It is possible to detect the curves one by one. Once there are three points whose plane leads to an accumulation value above a certain threshold t, all points lying on that plane can be removed from the input and hereby the detection efficiency be increased.

5. NEW ACCUMULATOR DESIGN

Without prior knowledge of the point cloud it is almost impossible to define proper accumulator arrays. An inappropriate accumulator, however, will lead to failures to detect some specific planes, difficulties in finding local maxima, display low accuracy, large storage space, and low speed. A trade-off has to be found between a coarse discretization that accurately detects planes and a small number of cells in the accumulator to decrease the time needed for the Hough Transform. Choosing a cell size that is too small might also lead to a harder detection of planes in noisy laser data.

5.1 Accumulator Array

For the standard implementation of the 2-dimensional Hough Transform (Duda and Hart (1971)) the Hough Space is divided into $N_\rho \times N_\varphi$ rectangular cells. The size of the cells is variable and can be chosen problem dependent. Using the same subdivision for the 3-dimensional Hough Space by dividing it into cuboid cells causes some major drawbacks.

A look at the structure of the accumulator in Fig. 1(a) when mapped to a ball explains the reason for this behavior. The ball represents all possible normal vectors that can define all different planes. Discretizing the surface of the ball with respect to φ and θ leads to different sized pieces. The blue colored latitude circles have the same distance while the longitude circles intersect at the poles. Therefore the cells closer to the poles are smaller and comprise less normal vectors. This means voting favors the larger equatorial cells. In plain terms this means that the points lying on a plane parallel to the xy-plane will vote for a larger number of small cells close to the poles, while points lying on a plane parallel to the xz-plane vote for a small number of larger cells. In the first case many cells will have an intermediate high value and are not as likely to be detected than a plane in the second case where a few cells have very high values.

5.2 Accumulator Cube

Censi and Carpin (2009) propose a design for an accumulator that is a trade-off between efficiency and ease of implementation. Their intention is to define correspondences between cells in the accumulator and small patches on the unit sphere with the requirement that the difference of size between the patches on the unit sphere is negligible. Their solution is to project the unit sphere $\mathbb{S}^2$ onto the smallest cube that contains the sphere using the diffeomorphism

$$\varphi\colon \mathbb{S}^2 \to \text{cube}, \quad s \mapsto s/\left\|s\right\|_\infty .$$

Each face of the cube is divided into a regular grid. Fig. 1(b) shows the resulting patches on the sphere. Given the normal vector of a plane the cell to be accumulated is calculated as follows. The side a_i of the cube is determined as the direction of the dominant coordinate n_d of the face normal. Scaling the normal vector with $1/n_d$ results in a projection onto the cube face, where the non-dominant coordinates of the normal vector transform into the cube coordinates $c_x = {}^{n_1}/_{n_d}$ and $c_y = {}^{n_2}/_{n_d}$. n_1 and n_2 denote the two non-dominant coordinates of the normal vector. c_x and c_y range between -1 and 1. Given the cube coordinates, the cell indices are then calculated as

$$a_{x/y} = \begin{cases} 1 & \dfrac{c_{x/y}+1}{2} = 1 \\ 1 + \text{nr\_cells} \cdot \dfrac{c_{x/y}+1}{2} & \text{else.} \end{cases}$$

This short insight into the mathematics shows that the transformation from $\mathbb{S}$ into accumulator indices and back into $\mathbb{S}$ is quite simple. The question remains whether the implementation is also efficient in terms of regularity of the cell sizes. Fig. 1(b)(a) shows the patches on the sphere. The regularity between all six cube faces is obvious. This means that in an environment that is composed of rectangularly arranged planes aligned with the coordinate system, all planes will be detected with the same probability due to the same sized accumulator cells. However, towards the corners of the cube faces the cells in the projection become smaller. The smaller the cells, the higher the number of cells, onto which the votes will be divided, accordingly less likelihood to be detected. This irregularity will always come into effect if the environment is rotated compared to the coordinate system of the accumulator. Due to the uneven matching of 8 corners to 6 faces the irregularities will always be of importance when considering rectangularly arranged environments as well as less structured environments.

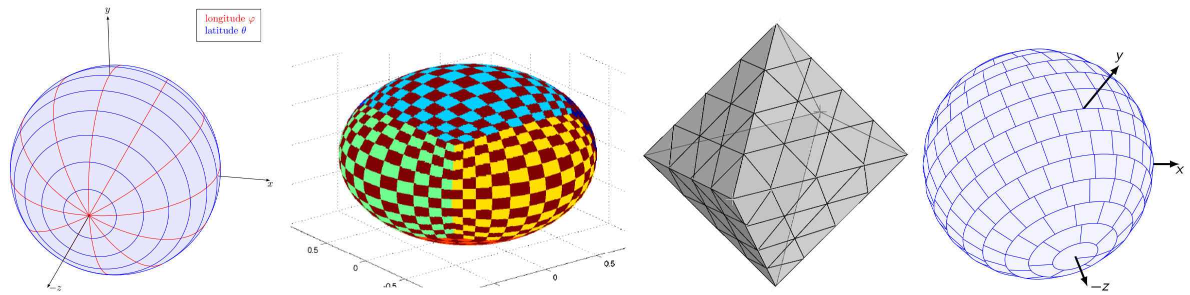
5.3 Polyhedral Accumulator

Zaharia and Preteux (2002) use the 3D Hough Transform for shape-based similarity retrieval. In this application geometric invariances play an important role as objects are aligned with the local coordinate system according to their principal axes. The same decomposition in the direction of all coordinate axes is achieved when partitioning the Hough Space by projecting the vertices of any regular polyhedron onto the unit space. The level of granularity can be varied by recursively subdividing each of the polyhedral faces. An example is given in Fig. 1(c). Each triangular face of the octahedron is divided into four triangles each of which can again be divided. It becomes obvious that each top of the octahedron has the same partitioning, i.e., the structure is invariant against rotation of 90° around any of the coordinate axes. This design brings along many advantages for the task of comparing objects that are aligned with respect to their principal axes. For detection of planes however it shows the same drawbacks as the cuboid design described in the previous section. When mapping the partitions back onto the unit sphere the patches will appear to be to be unequally sized.

5.4 Accumulator Ball

The three designs presented in the previous sections have one drawback in common, the irregularity between the patches on the unit sphere. While the simple array structure suffers from enormous differences between the patch sizes, the cuboid and octahedral designs reduce these differences drastically. A further benefit is their invariance against rotation of 90° around any of the coordinate axes. Problems remain with smaller rotations. If the planes to be detected do not align with the coordinate system the position of the plane decides about its likeliness to be detected. Still, the calculation of the cells to be accumulated is slightly more complicated.

We present a design for the accumulator with the intention of having the same patch size for each cell. To achieve this, the resolution in terms of polar coordinates has to be varied dependent on the position on the sphere. For this purpose the sphere is divided into slices. See Fig. 4 (left) for an illustration of the idea. The resolution of the longitude φ can be kept as for the accumulator array. φ' determines the distance between the latitude circles on the sphere, e.g., the thickness of the slices. Depending on the longitude of each of the latitude circles the discretization has to be adapted. One way of discretization is to calculate the step



(a) Array (b) Cube (c) Octahedron (d) Ball

Fig. 1. Mapping of the accumulator designs onto the unit sphere.

width θ' based on the size of the latitude circle at φ_i. The largest possible circle is the equator located at $\varphi = 0$. For the unit sphere it has the length $\max_l = 2\pi$. The length of the latitude circle in the middle of the segment located above φ_i is given by $\text{length}_i = 2\pi(\varphi_i + \varphi')$. The step width in θ direction for each slice is now computed as

$$\theta'_{\varphi_i} = \frac{360^\circ \cdot \max_l}{\text{length}_i \cdot N_\theta}.$$

The resulting design is illustrated in Fig. 1(d). The image clearly shows that all accumulator cells are of the same size. Compared to the previously explained accumulator designs, the accumulator cube and the polyhedral accumulator, a possible drawback becomes obvious when looking at the projections on the unit sphere. The proposed design lacks invariance against rotations of multiples of 90°. This leads to a distribution of the votes to several cells. In practice however this problem seems negligible since in most cases the planes to be detected do not align perfectly with the coordinate system. Experimental evaluation of the different accumulator designs in the following section support this claim.

6. EXPERIMENTAL EVALUATION

The different designs for storing the votes of the Hough Transform, are compared in this section using simulated as well as real laser scans. The simple array structure where φ and θ are discretized uniformly is the simplest way of discretizing the Hough Space. It is interesting to investigate whether the obvious flaws of this design come into effect in practical applications or if they are negligible. Second, out of the two designs that focus on symmetry with respect to the coordinate system we chose the cuboid design over the polyhedral design, since both designs seem to have similar characteristics and the cuboid design appears to be easier to manage. Third, the design of our accumulator ball with different discretization for each latitude slice of the unit sphere is evaluated against those other two designs.

6.1 Evaluation using Simulated Data

For easier evaluation we reduce the experimental setup to a simple test case. A cube with a side length of 400 cm is placed around the origin. Each side consists of 10000 points which are randomly distributed over the entire face of the cube with a maximal noise of 10 cm. Different rotations are applied to the cube to simulate different orientations of planes. The advantage of using this simple model is the existence of ground truth data for the actual planes. The cube possesses a perpendicular structure which is characteristic for most man-made indoor environments. As pointed out in Section 5 rotating the cube poses challenges to the accumulators as the sides are not symmetrically aligned with the coordinate axes anymore.

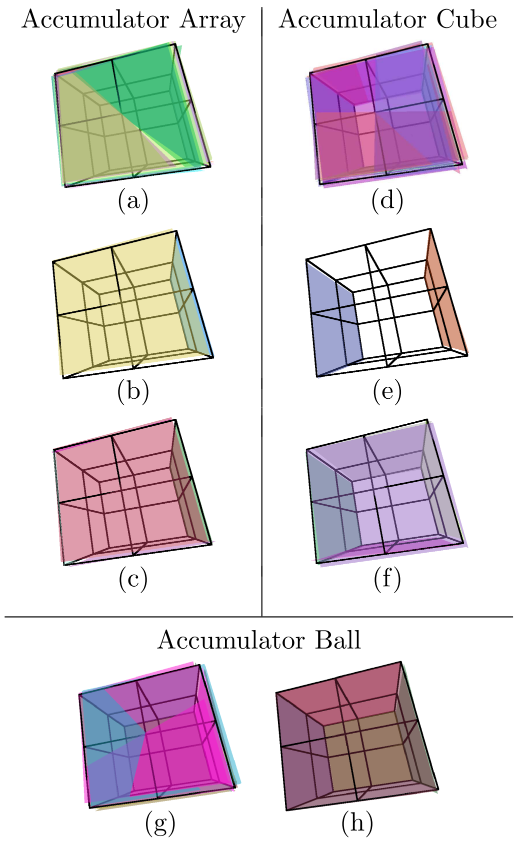


Fig. 3. Planes detected (SHT) using accumulator array (a)–(c), cube (d)–(f) and ball (g)–(h).
(a), (d), (g): 20 planes with the highest score.
(b), (e), (h): After peak search procedure, all planes with more than 90% of the maximal score.
(c), (f) The six planes with the highest score.

Quantitative Evaluation The first experiment presents a graphical investigation of the ability of the accumulator designs to correctly detect planes in the given point cloud. For this purpose we apply the SHT to the cube. To achieve comparable results the number of cells for each

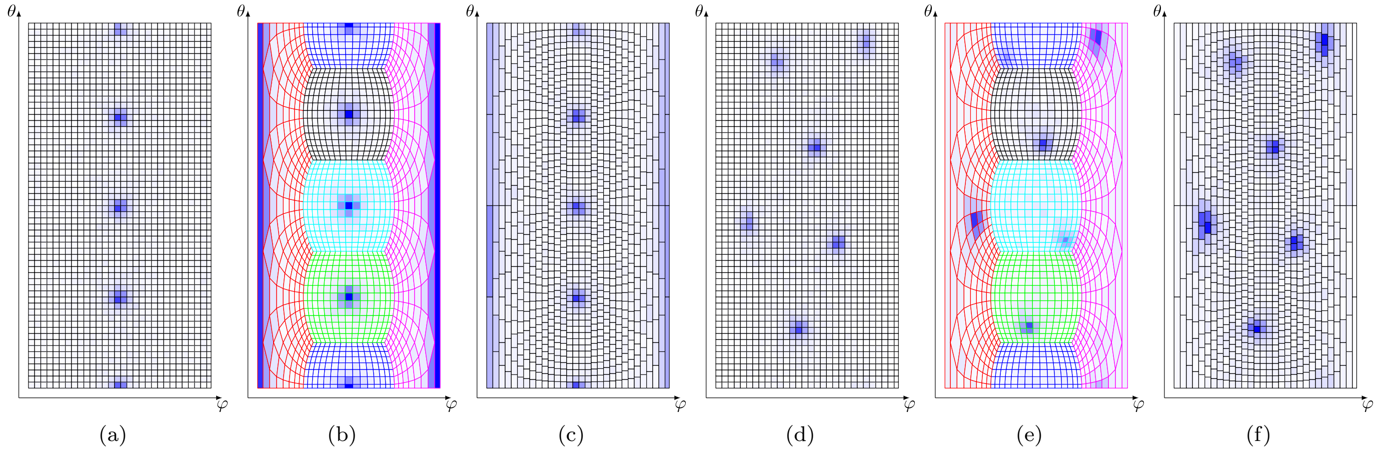


Fig. 2. Left to right: array, cube, ball design of the accumulator. The input data is the modeled cube rotated by $(0, 0, 0)$ (left) and $(45, 45, 45)$ (right).

accumulator needs to be approximately the same. $N_\rho = 100$ with a maximal distance of 600. For the accumulator ball we use $N_\varphi = 45$ and $N_\theta = 90$. This means that the slice around the equator consists of 90 patches. The total number of cells is 257100. The accumulator cube has a total number of 264600 cells when using 21×21 cells on each face. The simple accumulator is discretized with $N_\varphi = 38$ and $N_\theta = 76$ leading to 288800 cells.

In Fig. 2 the accumulators are plotted after applying 100000 iterations of the RHT. For better visibility only the slice with $198 < \rho < 204$ is shown, i.e., the distance that the planes have. The votes are drawn in blue, the darker a cell, the more votes it has received. The first three images show the results for the perfectly aligned cube and the cube rotated by $45\,^\circ$ around each axis. The images show that for the accumulator array the two planes corresponding to the highest and lowest φ values do not show up. For the ball design the peaks show up, but are not as high as the peaks around the equator. In practice this means that the several cells along the equator have higher votes as the cells around the poles and are therefore earlier detected as planes. The images (d)–(f) show the same experiments with the rotated cube. The six peaks that show up in the accumulator array vary in color. The same holds true for the cube. The peaks close to the corners of the cube (the faces of the accumulator are marked in different colors) show a lighter blue. These are the regions where the patches have the smallest size. The result is, that planes corresponding to those cells are less likely to be detected. For the ball design the peaks are most evenly colored in this scenario. This supports the previously mentioned assumption that the ball design is the best for detecting arbitrary planes due to its characteristic of having evenly sized patches in the Hough space.

For the next test case we use the laser scan model of the cube rotated by $(10, 10, 10)$ and apply the SHT to it using all three different accumulators. After the voting phase in the SHT, the peaks in the accumulator have to be found and a decision has to be made which of those peaks correspond to actual planes in the input data. As seen in Fig. 2 the votes for one plane spread over a small region in the Hough Space. To pick the best representation out of one of these regions we implemented a simple peak search strategy that is applied after voting. Starting in one corner of the accumulator, we run over the complete space with a small window, in this experiment $8 \times 8 \times 8$ cells. Within this window all values but the highest one are set to zero. This simple strategy might favor certain maxima but in our experiments this fact showed little influence on the results. For the accumulator cube we ran over each face separately. In the accumulator ball each slice consists of a different number of cells. This might cause problems when applying this windowed peak search. However, due to the small difference in size of two neighboring slices the covered region is not regularly shaped but still connected.

The resulting planes are shown in Fig. 3. The first image for each accumulator depicts the 20 planes with the highest score using no peak search strategy. Secondly, after applying the peak search strategy. For all accumulators all planes with a count up to $90\,\%$ of the highest score are considered to be actual planes. For the accumulator ball these planes are very close to the six faces of the cube model. For the accumulator cube only two planes are above the threshold. For the accumulator array the back and the bottom face are not among the top $90\,\%$ while the front appears three times. For the cube the threshold to correctly detect all six faces of the model is $88\,\%$. For the array all six faces of cube have a score above $80\,\%$.

Quantitative Evaluation The demands towards the HT are twofold. First, the planes need to be easily detected, i.e., each plane is represented by exactly one dominant maximum in the accumulator. In the example this means that each of the six highest peaks corresponds to one face of the cube. Second, the highest peak for each face is as close as possible to the ground truth of the plane. Fig. 4 shows an evaluation of the three different accumulator designs with respect to these aspects. On the left the ability to correctly detect all six planes is depicted. Nine different rotations are applied to the cube. The bars indicate the number of incorrectly detected planes, i.e., if the six highest peaks correspond to the six different faces of the cube, the error is zero. Each time the next highest peak corresponds to an already detected plane before every single face of the cube is represented by one peak, the error is incremented by one.

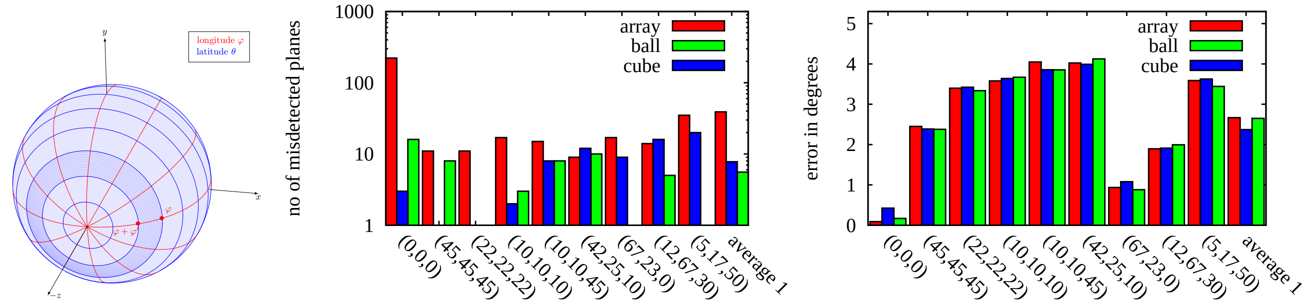


Fig. 4. Left: Calculation of the length of a longitude circle at φ_i. The segment in question is the darker colored one. Middle: Number of incorrectly detected planes on a logarithmic scale. Right: Angle error of the detected planes. Each set of bars represents one orientation of the cuboid laser scan model rotated by $(\alpha_x, \alpha_y, \alpha_z)$ as denoted underneath each set. α_x is the rotation around the x-axis, α_y around the y-axis and α_z around the z-axis

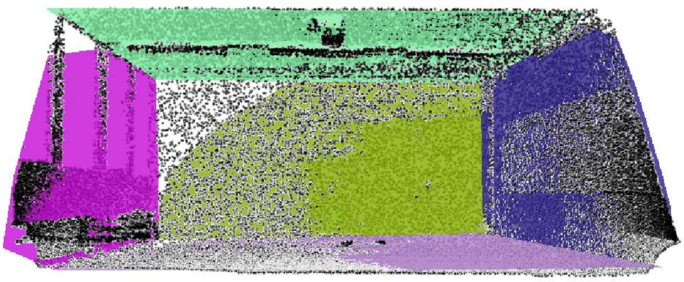


Fig. 5. Scan of an empty office with planes detected (RHT).

The results show clearly that the simple array structure has significant problems to correctly identify the six cube faces. It totally fails when the cube is aligned with the coordinate system, as motivated in Section 5. The uneven sizes of the patches lead to an uneven distribution of peaks against favor of the planes parallel to the xy-plane. This comes mostly in effect when the cube is aligned with the coordinate system, when rotated the effects diminish but do not disappear.

In Fig. 2 it became obvious that the accumulator ball has problems detecting planes that are parallel to the xy-plane. If the cap of the ball is divided into several patches, more cells intersect at the poles than at the other parts of the accumulator. This leads to distribution of the votes over all these patches, decreasing the count for each of these cells. This problem is solved by creating a circular cell around the pole that has the desired size of the patches and proceed with the rest of the sphere in the same manner as before. Using this design the ball and the cube show similar performance. The ball performs better in some test cases, the cube in others. On average the ball design slightly outperforms the cube.

The chart on the right of Fig. 4 shows the sum of the angle errors for the detected planes. For each side of the cube the cell with the highest vote is used and the angle between the normal vector and the ground truth normal vector of the plane is calculated as error. The results show only small, negligible differences between the different accumulator designs. This indicates that once a plane is detected correctly the parameters are calculated equally well with each accumulator design.

6.2 *Evaluation using Real Laser Data*

To show the applicability of the Hough Transform to real laser data, we apply the RHT to a laser scan of an empty office. The result is shown in Fig. 5. All planes were correctly detected within less than 600 ms on an Intel Core 2 Duo 2.0 GHz processor with 4 GB RAM. For further evaluation on the applicability of the Hough Transform see Borrmann and Elseberg (2009).

7. CONCLUSION

In this paper we proposed a novel accumulator design, whose cells are of equal size. This property leads to an easier detection of planes when using the Hough Transform, since maxima are pronounced equally independent of their position in the accumulator. The applicability of the design for the detection of planes in simulated as well as in real laser scans was shown by the presented experiments. We also compared our approach to previously used designs.

REFERENCES

Bauer, U. and Polthier, K. (2008). Detection of Planar Regions in Volume Data for Topology Optimization. *Lecture Notes in Computer Science*.

Borrmann, D. and Elseberg, J. (2009). Deforming Scans for Improving the Map Quality Using Plane Extraction and Thin Plate Splines. Master's thesis, Univ. Osnabrück.

Censi, A. and Carpin, S. (2009). HSM3D: Feature-Less Global 6DOF Scan-Matching in the Hough/Radon Domain. In *Proc. ICRA*.

Duda, R.O. and Hart, P.E. (1971). Use of the Hough Transformation to Detect Lines and Curves in Pictures. Tech. Note 36, AI Center, SRI International.

Fischler, M.A. and Bolles, R.C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Comm. of the ACM*, 24, 381 – 395.

Hough, P.V.C. (1962). Method and Means for Recognizing Complex Patterns. US Patent 3069654.

Kiryati, N., Eldar, Y., and Bruckstein, A. (1991). A Probabilistic Hough Transform. *Pattern Recognit.*, 24(4), 303–316.

Schnabel, R., Wahl, R., and Klein, R. (2007). Efficient RANSAC for Point-Cloud Shape Detection. *Comput. Graphics Forum*.

Xu, L., Oja, E., and Kultanen, P. (1990). A new Curve Detection Method: Randomized Hough Transform (RHT). *Pattern Recognit. Lett.*, 11, 331–338.

Zaharia, T. and Preteux, F. (2002). Shape-based Retrieval of 3D Mesh Models. In *Proc. ICME*.