Real-Time Detection of Planar Regions in Unorganized Point Clouds

Frederico A. Limberger[1]**, Manuel M. Oliveira**[1] **(Advisor)**

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)

{falimberger|oliveira}@inf.ufrgs.br

Abstract. *Automatic detection of planar regions in point clouds is an important step for many graphics, image processing, and computer vision applications. While laser scanners and digital photography have allowed us to capture increasingly larger datasets, previous approaches for planar region detection are computationally expensive, precluding their use in real-time applications. We present an $O(n \log n)$ technique for plane detection in unorganized point clouds based on an efficient Hough-transform voting scheme. It works by clustering sets of approximately co-planar points and by casting votes for these clusters on a spherical accumulator using a trivariate Gaussian kernel. A comparison with competing techniques shows that our approach is considerably faster and scales significantly better than previous ones, being the first practical solution for deterministic plane detection in large unorganized point clouds.*

1. Introduction

Automatic plane detection in point clouds is a key component in many graphics, image processing, and computer vision applications. These include, among others, model reconstruction for reverse engineering, camera calibration, object recognition, augmented reality, and segmentation. The recent popularization of laser scanners has led to an increasingly growth in the sizes of the available datasets, and point clouds containing tens of millions of samples are now commonplace. Software applications like *SynthExport* [Hausner 2010] and *Photosynth* [Photosynth 2008] also allow us to extract point clouds from large collections of digital images. However, previous techniques for detecting planar regions in point clouds are computationally expensive and do not scale well with the size of the datasets. For performance improvement, they often exploit non-deterministic strategies, such as working on a randomly-selected sub-set of the original samples. While this can reduce execution time, these techniques are still unable to achieve real-time performance even on datasets containing just tens of thousands of points. More importantly, their results depend on the selected sample sub-sets and, therefore, there is no guarantee that all relevant planes will be detected, or that such results will be consistent across multiple executions.

We present an efficient technique to perform deterministic plane detection in unorganized point clouds. Our approach scales well with the size of the datasets, is robust to the presence of noise, and handles point clouds with different characteristics in terms of dimensions and sampling distributions. While the actual running times depend on specific characteristics of the dataset (*e.g.*, the number of planar regions and noise level), our technique is several orders of magnitude faster than previous ones.

Our approach is based on a robust and fast algorithm to segment point clouds into approximately planar patches, even in the presence of noise or irregularly distributed samples. We then use the identified sample clusters to obtain an efficient Hough-transform voting scheme by casting votes for each of these clusters (instead of for individual samples) on a spherical accumulator. For voting, we use a Gaussian kernel obtained from the cluster's covariance matrix and centered at the cluster's best fitting plane. In this sense, our approach extends the kernel-based voting scheme proposed by Fernandes and Oliveira [Fernandes and Oliveira 2008] using a trivariate Gaussian distribution defined over spherical coordinates (θ, ϕ, ρ). While plane detection in unorganized point clouds might seem as an immediate extension of line detection in images, the lack of explicit neighborhood information among samples imposes significant challenges, requiring new clustering and accumulation-management strategies.

An article describing our contributions, entitled *Real-Time Detection of Planar Regions in Unorganized Point Clouds* [Limberger and Oliveira 2015a], was published in *Pattern Recognition* journal, one of the most important in the field.

2. Efficient Plane Detection in Point Clouds

The Standard Hough transform (SHT) for plane detection iterates over each sample in the point cloud casting votes in a 3-D accumulator for all possible planes passing through that sample [Hough 1962, Duda and Hart 1972]. Although deterministic, this brute-force strategy results in a high computational cost. The optimizations introduced in this thesis allow a software implementation to operate in real-time for sufficiently large point clouds (up to 10^5 samples) using current hardware.

2.1. Clustering of Approximately Coplanar Samples

Clustering of approximately coplanar samples is key to our technique as it optimizes the voting procedure, which is the Hough transform's bottleneck. For efficiency, we perform clustering through spatial subdivision using an octree. This has proven to be a good choice both in terms of efficiency and quality of the results.

The clustering procedure starts at the root node, which includes the entire point cloud. At each octree node, the procedure checks for approximate coplanarity among its samples using principal component analysis (PCA). If the test is successful, the node is marked as an approximately coplanar cluster and the subdivision process stops for that node. If the number of samples inside the node is smaller than a given threshold, the node is marked as not containing a coplanar cluster and it is no further refined. Otherwise, the subdivision process continues. Figure 1 illustrates the octree-based clustering process.

2.2. Computing Gaussian Trivariate Kernels for Cluster Voting

We use a Gaussian trivariate kernel to cast votes for a cluster on the 3-D spherical accumulator. To obtain such a kernel, we compute the cluster's covariance matrix $\Sigma_{(\theta,\phi,\rho)}$ in the (θ, ϕ, ρ) spherical space from its covariance matrix $\Sigma_{(x,y,z)}$ in the (X, Y, Z) Euclidean space. Using first-order uncertainty propagation analysis, $\Sigma_{(\theta,\phi,\rho)} = J\Sigma_{(x,y,z)}J^T$, where J is the Jacobian matrix of the transformation:

$$J = \begin{pmatrix} \frac{\partial \rho}{\partial p_x} & \frac{\partial \rho}{\partial p_y} & \frac{\partial \rho}{\partial p_z} \\ \frac{\partial \phi}{\partial p_x} & \frac{\partial \phi}{\partial p_y} & \frac{\partial \phi}{\partial p_z} \\ \frac{\partial \theta}{\partial p_x} & \frac{\partial \theta}{\partial p_y} & \frac{\partial \theta}{\partial p_z} \end{pmatrix}.$$

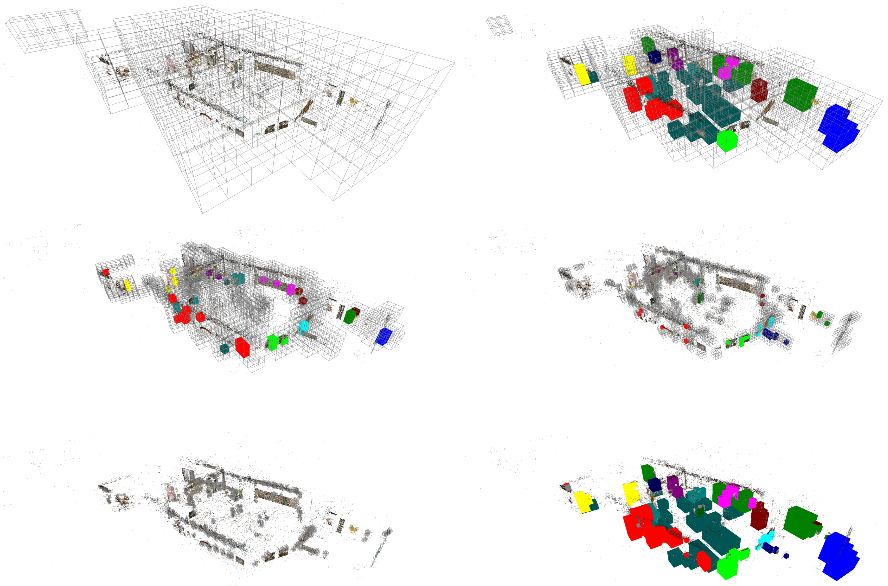

Figure 1. Adaptive octree refinement and sample clustering for the Museum dataset. From left to right, top to bottom, the first five images show the 6th, 7th, 8th, 9th, and 10th levels of the octree. The image at the bottom right shows all nodes at different octree levels containing coplanar samples. Note that once a planar patch is found the subdivision stops for that branch. Each color represents one detected plane, whose reconstructions are shown in Fig. 3.

The Gaussian kernel used for voring for a given sample cluster is then obtained as:

$$G(\mathbf{x}|\mu, \Sigma_{(\theta,\phi,\rho)}) = \frac{1}{(2\pi)^{3/2}|\Sigma_{(\theta,\phi,\rho)}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^t \Sigma^{-1}_{(\theta,\phi,\rho)}(\mathbf{x}-\mu)\right), \qquad (1)$$

where $|\Sigma_{(\theta,\phi,\rho)}|$ is the determinant of $\Sigma_{(\theta,\phi,\rho)}$, and μ and $\mathbf{x}$ are, respectively, the (θ, ϕ, ρ) coordinates of the cluster's best-fitting plane and of an accumulator cell receiving votes.

2.3. Cluster Voting using 3D Gaussian Distributions

Given the Gaussian distribution in Eq. (1), we cast votes on an adapted version of the *ball accumulator* of Borrmann et al. [Borrmann et al. 2011]. The number of votes cast by a plane on the cells of a spherical accumulator decreases as one moves from the equator to the poles. This is illustrated in Fig. 2 (left), which compares the distribution of votes cast by a cluster as it is rotated around the origin. Note that the number of votes cast on cells around the poles are significantly smaller than the ones near the equator. Fig. 2 (center) shows two instances of the rotated point cloud: one near the north pole and the other near the equator. The noise in the point cloud lends to some uncertainty on the plane's orientation, which is represented by a cone of normals around the normal of the best-fitting plane (shown in red). On the equator, such uncertainty causes some votes to be cast in a small θ and ϕ neighborhood around the (θ, ϕ, ρ) coordinates of the best fitting plane. There, equal angular steps in θ and in ϕ correspond to arc lengths of equal sizes, resulting in an isotropic Gaussian kernel in the (θ, ϕ) subspace. Such a Gaussian is illustrated on the top portion of Fig. 2 (right). Near a pole, the uncertainty on the plane's

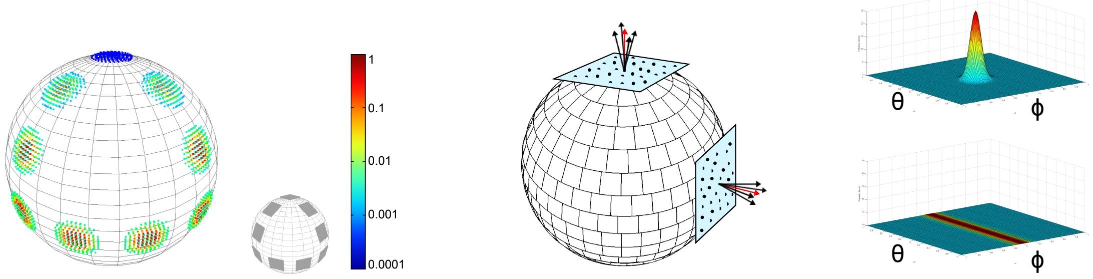

Figure 2. The number of votes cast by a cluster as it is rotated varies with the position on the spherical accumulator (left). The color scale indicates the number of votes, while the thumbnail image on its right shows the best-fitting planes corresponding to the rotated clusters. (center) On the equator, the uncertainty on the plane orientation lends to votes on a small isotropic neighborhood in the (θ, ϕ) subspace. At (next to) a pole, the same uncertainty on the plane orientation lends to a small uncertainty in the ϕ dimension, but to a big uncertainty in the θ dimension, as θ can range from 0 to 360 degrees. (right) isotropic (top) and truncated anisotropic (bottom) Gaussian kernels in the (θ, ϕ) subspace associated to the cluster near the equator and near the pole, respectively.

normal lends to a small uncertainty in the parameter ϕ, but to a huge uncertainty in the parameter θ, as at the pole the value of θ varies from 0 to 360 degrees. This results in a highly anisotropic Gaussian kernel in the (θ, ϕ) subspace, as shown by the truncated kernel at the bottom of Fig. 2 (right). This explains the smaller number of votes per cell as a cluster approaches a pole.

2.4. Peak Detection

The last stage of a Hough-transform consists of detecting peaks of votes in the accumulator, which are interpreted as the detected planes. We apply a low-pass filter to the accumulator to smooth the voting map, consolidating adjacent peaks. As discussed in the previous sub-section, for the same relative sample distribution, the amount of votes cast to individual accumulator cells near a pole tends to be smaller than the amount of votes cast to cells in other regions of the accumulator (Fig. 2 (left)). Thus, the list of detected planes is sorted in decreasing order based on the sum of the weights (*i.e.*, importance) of all clusters that voted for each plane. The importance of a cluster is computed based on its number of samples and on its spatial coverage relative to the entire point cloud.

3. Results

To evaluate the efficiency and effectiveness of our approach, we have tested our technique on different types of point clouds, varying in size (number of points), sampling density, and complexity (number of underlying planes). The point clouds were extracted from sets of photographs using SynthExport and Photosynth, except *Computer* which is from [Borrmann et al. 2011]. To evaluate its accuracy, we created a synthetic dataset by sampling the faces of a cube containing around 1 million points and 2.5% of uniformly-distributed noise. To evaluate robustness to missing samples and noise, we downsampled this dataset to 48,000 points and added 1% of Gaussian noise. For this experiment, each face of the cube corresponds to three discontinuous stripes of samples covering approximately 60% of its original area. We then detected planes in both versions of the cube after rotating them by arbitrary amounts and around arbitrary axes. In all cases, our technique accurately detected the six planes of the cube.

We compared the performance of our technique to the state-of-the-art approaches for plane detection in point clouds: the optimized RANSAC of Schnabel et al. [Schnabel et al. 2007], the RANSAC implementation for plane detection in point clouds available in the *Point Cloud Library* (PCL) v1.7 [Rusu and Cousins 2011], and to Borrmann et al.'s [Borrmann et al. 2011] implementation of the Randomized Hough Transform (RHT) (an efficient non-deterministic Hough transform). These implementations proved to be the most efficient ones for plane detection using RANSAC and RHT, respectively. All experiments were performed on an Intel i7-2600 3.4 GHz CPU with 16 GB of RAM. Reported results are shown in Table 1. They reveal that our approach (KHT) processes the *Computer* dataset with its 68 thousand samples in approximately 22 milliseconds, and the *Museum* dataset, which contains 179 thousand samples, in 25 milliseconds. For larger point clouds (e.g., *Bremen*, with 20 million samples) the octree creation dominates the execution time of our method. Still, our technique processes the Bremen dataset 353 faster than Schnabel et al.'s RANSAC technique [Schnabel et al. 2007], and 3,586 times faster than PCL's RANSAC [Rusu and Cousins 2011]. The available implementation of the RHT could not handle the entire dataset. Working on the full dataset, our technique is still 20 times faster than RHT working on a subset containing only 10% of the original samples. The most representative planes detected by our method on each dataset can be seen in Fig. 3.

Table 1. Performance comparison of our approach (3D KHT) against RANSAC and RHT for various datasets. The entries of the table show the execution times (in seconds) of the three techniques for these datasets. (*) The RHT was computed with a simplified version of Bremen dataset containing only 2 million samples, because the available implementation did not support larger inputs.

	Computer	Room	Utrecht	Museum	Bremen
Point Cloud Size	68 852	112 586	160 256	179 744	20 332 246
3D KHT	0.022	0.041	0.040	0.025	2.105
RANSAC [Schnabel et al. 2007]	0.340	0.774	0.919	1.200	745.055
RANSAC [Rusu and Cousins 2011]	0.424	3.293	15.412	302.610	7,531.010
RHT	0.121	6.313	2.814	11.96	42.824 *

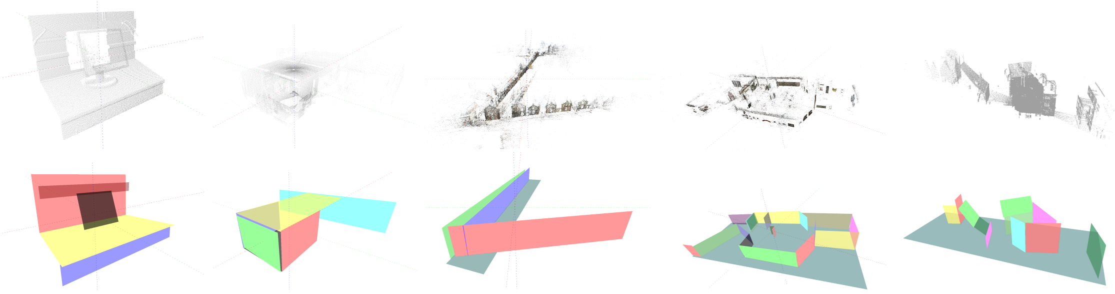

Figure 3. Datasets used for performance comparisons. Point clouds (top) and the most representative planes detected by our technique (bottom). From left to right, the datasets are: *Computer*, *Room*, *Utrecht*, *Museum*, and *Bremen*.

4. Conclusion

This work presented an efficient Hough-transform technique to perform deterministic plane detection in unorganized point clouds. Our approach uses a fast and robust algorithm to segment clusters of approximately coplanar samples, and casts votes for individual clusters, instead of for individual samples, on a spherical accumulator. For this, we use a trivariate Gaussian kernel that models the uncertainty about the position and orientation of the plane represented by the cluster.

While previous approaches for plane detection have basically resorted to randomly selecting a subset of the samples as a way to reduce execution time, we have undertaken the more fundamental strategy of designing an efficient algorithm with lower asymptotic cost. Our experiments have shown that our approach is several orders of magnitude faster than existing (non-deterministic) techniques for plane detection in point clouds, such as RANSAC and RHT, and scales better with the size of the datasets. It is also robust to noise, and handles point clouds with different characteristics in terms of dimensions and sampling distributions. As such, our technique provides the first practical solution for deterministic plane detection in large unorganized point clouds. For more information about the method and the source code we refer the reader to [Limberger and Oliveira 2015b].

Acknowledgments This work was sponsored by CNPq (grants 131002/2012-0, 482271/2012-4, and 308936/2010-8), and FAPERGS PQG 10/1322-0.

References

Borrmann, D., Elseberg, J., Lingemann, K., and Nüchter, A. (2011). The 3D Hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Research*, 2(2):32:1–32:13.

Duda, R. O. and Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15.

Fernandes, L. A. F. and Oliveira, M. M. (2008). Real-time line detection through an improved Hough transform voting scheme. *Pattern Recognition*, 41(1):299–314.

Hausner, C. (2010). Synthexport v. 1.1.0. `http://synthexport.codeplex.com/`.

Hough, P. (1962). Method and means for recognizing complex patterns.

Limberger, F. A. and Oliveira, M. M. (2015a). Real-time detection of planar regions in unorganized point clouds. *Pattern Recognition*, 48(6):2043 – 2053.

Limberger, F. A. and Oliveira, M. M. (2015b). Real-time detection of planar regions in unorganized point clouds. `http://www.inf.ufrgs.br/~oliveira/pubs_files/HT3D/HT3D_page.html`.

Photosynth (2008). *Photosynth.* Microsoft. https://photosynth.net/.

Rusu, R. and Cousins, S. (2011). 3d is here: Point cloud library (pcl). In *Proceedings of Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4.

Schnabel, R., Wahl, R., and Klein, R. (2007). Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226.