D-KHT: Real Time Plane Detection in Depth Images

Eduardo Vera Sousa
Instituto de Computação
Universidade Federal Fluminense
Niterói, RJ 24210–346
Email: eduardovera@ic.uff.br

Luiz Velho
Instituto Nacional de
Matemática Pura e Aplicada
Rio de Janeiro, RJ 22460–320
Email: lvelho@impa.br

Leandro A. F. Fernandes
Instituto de Computação
Universidade Federal Fluminense
Niterói, RJ 24210–346
Email: laffernandes@ic.uff.br

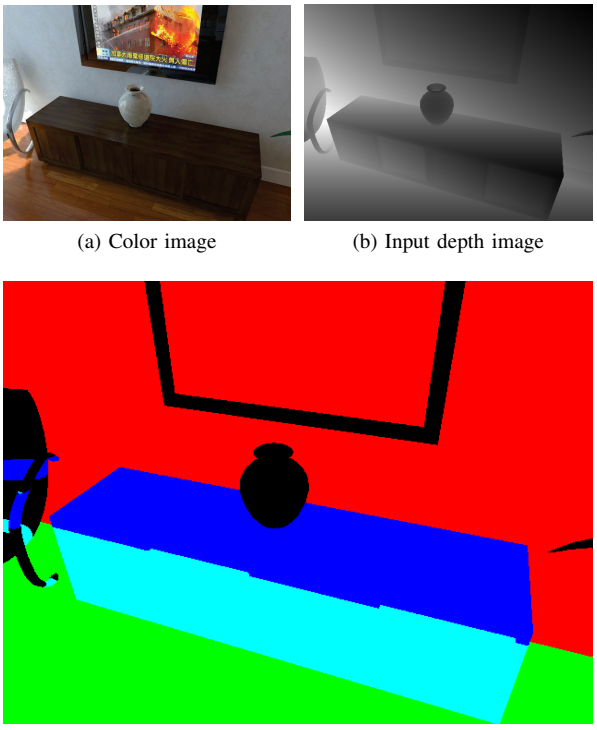
***Abstract*—The automatic detection of geometric primitives (*e.g.* planes, spheres, etc.) in depth images provides the basis for solving many computer vision problems. The applications range from robotics to augmented reality. For plane detection, the quality of previous techniques is strongly related to the amount of noise and discontinuities or, in the case of unorganized point clouds, depends on complex structures to organize the points, besides having high computational cost. In this paper, we present a real-time deterministic algorithm for plane detection in depth images. By using an implicit quadtree to cluster approximately coplanar points in the 2.5-D space associated with an efficient Hough Transform voting scheme and a hill climbing strategy to find local maxima, we are able to reach real-time detection. Experiments show that our approach works well even in presence of noise, partial occlusion, and discontinuities.**

I. Introduction

Mobile devices have become a cheap and universal technology. Depth sensors attached to these handheld computers, like Structure Sensor [1] and Project Tango [2], allow obtaining spatial data from the environment around it in real-time and motivates the development of real-time variants of classic computer vision algorithms to handle these data. The detection of planar structures has a lot of practical applications like image-based scene reconstruction, camera calibration, autonomous vehicle navigation and augmented reality, for example. State-of-the-art plane detection techniques that use unorganized point clouds are not appropriate for depth images by requiring special data structures to handle points in 3-D space or by using non-deterministic approaches to reduce the amount of processing. Region growing-based techniques, that segment planar patches, on the other hand, are designed to the regular structure of depth data. Its quality, however, is inversely proportional to the amount of noise and presence of occlusions or missing depth information in order to avoid multiple detections of the same plane.

The Hough Transform (HT) [3] is a classical algorithm for detection of geometric primitives. In its classical version, it consists in evaluations of all input points in a n-dimensional space and in the identification of all instances of the geometric primitive sought which could contain each evaluated input point. Then, each found instance votes in a space expanded by the parameters of the analytic representation of the geometric primitive sought. At the end of this step, local maxima in this parameter space correspond to the most likely instances in the original n-dimensional space. The classical form of the HT, besides having high memory footprint, has a brute force-based voting scheme, evaluating each input individually, which makes of it a computationally expensive approach. The Kernel-based Hough Transform (KHT) [4], by clustering the input points and performing an efficient voting scheme, is able to run in real-time and with low memory footprint.

In this work[1], we present an efficient technique for plane de-



(a) Color image (b) Input depth image

(c) Detected planes

Fig. 1. Outcome of the presented technique (best viewed in color): (a) Color image of the scene; (b) Input depth image with 640×480 pixels and 16-bits of depth; and (c) Detected planes. Each detected plane was assigned to a color. Note that the approach works well, even on presence of non-planar surfaces, represented by black pixels in (c).

[1]Master's Thesis presented to Computing Graduate Program of Universidade Federal Fluminense

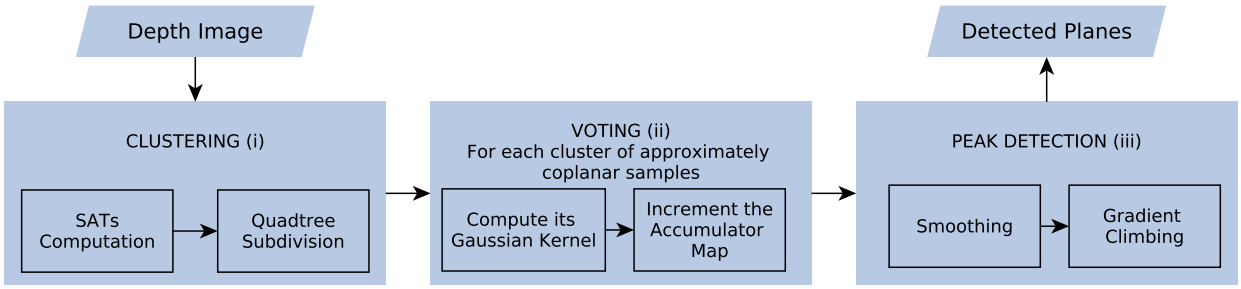


Fig. 2. The D-KHT pipeline: (i) Considering a depth image as input, the first step is compute SATs to efficiently get the covariance matrix and the mean point inside a cluster during the clustering step, which is done by using an implicit quadtree to identify rectangular image regions having approximately coplanar points in 3-D. (ii) For each region identified in step (i), the Gaussian distribution which describes the uncertainty of the best-fitting plane for the cluster is calculated and used to increment the spherical accumulators map. (iii) Then, we use a hill climbing scheme to find local maxima in the accumulators map, whose coordinates correspond to the most likely planes in the image.

tection in depth images, based on the HT and implementing the efficient kernel-based voting scheme of KHT [4]. **The main contribution of this work is an approach to plane detection in depth images, the Depth Kernel-based Hough Transform (D-KHT), which has asymptotic cost $O(n)$, and is from ~ 3 to ~ 5 times faster than state-of-the-art techniques**. The deterministic solution allows the implementation of the algorithm for real-time plane detection, with low memory footprint. Besides that, the presented approach is robust to the typical noise found in this type of image and to multiple detections of the same plane, even with occlusions and missing portions of depth information. Thus, the challenges of this work include:

- By using the regular structure and the capture model of depth images as restrictions, show that we are able to reduce the processing time required for the clustering step of Kernel-based Hough Transform;
- By using the observation that local maxima in the parameter space of Hough Transform are typically close to parameters of mean plane of each cluster, show that we are able to reduce processing time required for finding local maxima on voting map (*i.e.*, most likely planes); and
- Provide an algorithm for plane detection with quality and execution time that exceeds the state-of-the-art approaches.

The presented algorithm is one of the products resulting from the development of a major project for the real-time approximation of the geometry of indoor environments by planes by using depth sensors attached to mobile devices. Details of this work are described in [5].

Fig. 1c shows the result obtained by the proposed algorithm applied to the depth image in Fig. 1b. Each color in Fig. 1c represents a plane detected by our algorithm in 15.34 ms in a PC with a 3.4 GHz processor. Regions in black are non-planar surfaces identified by the technique. The input image (Fig. 1b) has 640×480 pixels with 16 bits in depth. The image corresponds to frame 294 of the *Living Room* [6] dataset.

Notice that, given the space restrictions, this paper does not cover all the details or results of the approach presented in the Master's Thesis manuscript. For a better understanding, please refer to the full work [5] and to the paper submitted to Pattern Recognition Letters. Both are attached to this submission.

II. Related Works

The Hough Transform (HT) [3] is one of the most popular techniques for primitives detection. It is commonly applied in low dimensional spaces, considering that its complexity increases with the space dimensionality. The HT for straight line detection uses the parameters of the normal equation of the line to expand a 2-D space, discretized as an accumulators map. The discretization step adopted is strongly related to detection quality and processing time. Given an image, for each edge pixel, the set of lines that may pass through it is evaluated and the bin of the accumulators map correspondent to the parameters of each line is increased by *one*. At the end of this process, local maxima generated in this accumulators map address the parameters of the most-likely lines in the image. Notice that the voting process is a brute force algorithm, *i.e.* have high computational cost, considering that each edge pixel vote individually in the accumulators map.

Fernandes and Oliveira [4] proposed an efficient HT voting scheme to avoid voting by brute force. This method, the Kernel-based Hough Transform (KHT), is based on clustering approximately collinear edge pixels and vote for the entire cluster considering the Gaussian uncertainty associated with the straight line that best explains the cluster, *i.e.*, each cluster cast votes under a Gaussian distribution in the accumulators map. Following the KHT efficient voting scheme, Limberger and Oliveira [7] proposed the 3-D KHT for plane detection in unorganized point clouds in 3-D spaces. This technique requires a spatial structure to organize the point cloud before starting the voting step. For this purpose, in the clustering step, they used an octree where each leaf node has only approximately coplanar points or outliers. Latter will be ignored in the voting step.

Surface Growing (SG) [8, 9] is a region-growing-based technique more appropriate for plane detection in depth images. It detects planar patches by expanding the area around a seed pixel while the normal vector of the surface keeps smooth. SG works well in the presence of a small amount of noise but, on the other hand, has high computational cost and requires

connectivity between pixels to avoid multiple detections of the same plane. Depth images with missing depth information, occlusion or high noise rate may lead to planes misdetection.

III. Depth Kernel-based Hough Transform

An overview of the presented technique may be seen in Fig. 2, which illustrates the main idea in each step of the technique: (i) considering a depth image as input, our approach uses Summed-Area Tables (SATs) [10] and an implicit quadtree [11] to efficiently clusterize neighbour pixels as nodes having approximately coplanar points, in 3-D; (ii) for each cluster, we use first-order error propagation [12] to estimate the mean plane and its associated Gaussian uncertainty; then, we compute a trivariate-Gaussian kernel, which guides the voting process by defining the increment of the accumulator bins around the parameters of the mean plane; and (iii) after a smoothing step to consolidate adjacent peaks of votes, the location of local maxima correspond to the parameters of the most likely planes in the depth image.

A. Clustering

The proposed technique is based on clustering the pixels from input depth image (Fig 2, step i). Thus, instead of considering each pixel as an individual input, it's possible to reduce the computational cost of the voting process by considering clusters of pixels with similar features. Fernandes and Oliveira [4] considered line segments as clusters. Limberger and Oliveira [7] considered as clusters the nodes of an octree, which is a structure to subdivide the 3-D space. Considering the presented approach, it's known that the input image depth does not have overlapping points along the z-axis. Thus, we used an implicit quadtree defined over the image and recursively subdivide its nodes whenever the approximate coplanarity is not observed in the points belonging to this node. To consider a node as having approximately coplanar points, one must obtain

$$\mu_{(x,y,z)} = \begin{pmatrix} \mu_x \\ \mu_y \\ \mu_z \end{pmatrix} \text{ and } \Sigma_{(x,y,z)} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{pmatrix}, \quad (1)$$

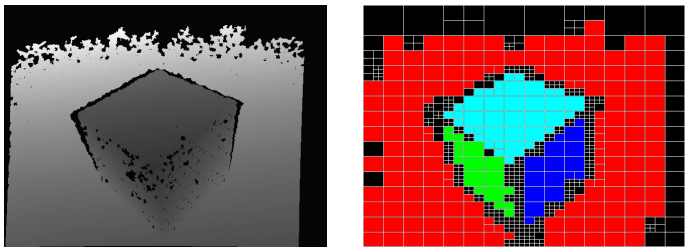
where $\mu_{(x,y,z)}$ corresponds to the node mean point and $\Sigma_{(x,y,z)}$ corresponds to the covariance matrix of the points, and check if $2\sqrt{\lambda_1} < s_t$, where λ_1 is the least eigenvalue of $\Sigma_{(x,y,z)}$ and s_t is a threshold for the accepted thickness of a node.

The covariance between two random variables a and b is given by

$$\sigma_{ab} = \frac{1}{m-1}\left(\sum_{k=1}^{m} a_k\, b_k - \mu_b \sum_{k=1}^{m} a_k - \mu_a \sum_{k=1}^{m} b_k + m\,\mu_a\,\mu_b\right), \quad (2)$$

where

$$\mu_a = \frac{1}{m}\sum_{k=1}^{m} a_k, \quad \mu_b = \frac{1}{m}\sum_{k=1}^{m} b_k, \quad (3)$$



(a) Input depth image (b) Clustered input

Fig. 3. The Cube dataset is comprised of a cube over a flat surface (a). The depth image was acquired using a Structure Sensor. Black pixels in (a) corresponds to missing depth information. Image (b) shows the leaf nodes of the quadtree built over the input image. Nodes identified by the same color corresponds to clusters of point that voted to the same detected plane. Black nodes in (b) corresponds to outliers and are not considered for voting procedure.

and m is the number of samples. For the computation of $\Sigma_{(x,y,z)}$, m is the amount of points inside a node and $\{a_k\}_{k=1}^{m}$ and $\{b_k\}_{k=1}^{m}$ are the x, y and z coordinates of each point. One shall notice that these coordinates are in the unit of measurement used by the device and the pixel coordinates must be converted to it before computing $\mu_{(x,y,z)}$ and $\Sigma_{(x,y,z)}$ [5].

By taking advantage of the regular structure of the discrete depth image, it is possible to apply SATs in covariance matrix and mean point calculation in order to reduce execution time. In practice, each summation in Eq. 2 and Eq. 3 become four access to a SAT previously calculated, leading to a constant time in covariance matrix and mean point calculation. In total, nine SATs are needed. The concept of SATs and its applications in the estimation of approximate coplanarity are better described, respectively, in Chapters 3 and 4 of [5].

B. Voting

The parameter space used in the presented approach is expanded by the parameters θ, ϕ and ρ of the normal equation of the plane, given by:

$$\rho = x\,\sin\phi\,\cos\theta + y\,\sin\phi\,\sin\theta + z\,\cos\phi, \quad (4)$$

where x, y and z are the coordinates of a point that belongs to the plane. In the presented approach, the type of accumulators map used the voting process (Fig. 2, step ii) is the one introduced by Borrmann *et al.* [13], *i.e.*, an spherical accumulators map, whose main property is the fact that its cells are about the same size, so that the same weight is given to all accumulators, regardless of their distances from the poles. The model of capture of depth images assumes the camera center as the origin of the 3-D space and its principal axis aligned to z-axis of space. Thus, all the depth values in the image has positive z values. The parameters space is restricted to $\theta \in [-\pi, +\pi)$, corresponding to the elevation angle of the accumulators map; $\phi \in [0, \pi)$, as the azimuth; and $\rho \in [0, \rho_{high}]$, where ρ_{high} is the camera distance to the farthest point in the image. The discretization step is guided by the parameters N_θ and N_ρ, which concerns to the a number of bins along the directions θ

and ρ, respectively. The number of bins along the ϕ direction is given as a function of the elevation angle.

In the clustering step (Section III-A), we obtained the covariance matrix, $\Sigma_{(x,y,z)}$, and the mean point, $\mu_{(x,y,z)}$, for each node having approximately coplanar points. Let $\overrightarrow{v_1}$ be the eigenvector associated to the least eigenvalue of $\Sigma_{(x,y,z)}$, λ_1. Notice that $\overrightarrow{v_1}$ is the normal vector $\overrightarrow{n} = (n_x, n_y, n_z)^T$ of the plane that better explains the points in a cluster. Thus, we are able to retrieve the mean plane parameters as

$$\mu_{(\rho,\phi,\theta)} = \begin{pmatrix} \mu_\rho \\ \mu_\phi \\ \mu_\theta \end{pmatrix} = \begin{pmatrix} n_x\,\mu_x + n_y\,\mu_y + n_z\,\mu_z \\ \cos^{-1}(n_z) \\ \tan^{-1}\left(\frac{n_y}{n_x}\right) \end{pmatrix}. \quad (5)$$

Our trivariate-Gaussian distribution will be centered at the parameters of $\mu_{(\rho,\phi,\theta)}$. The covariance in parameters space is obtained by first-order error propagation as

$$\Sigma_{(\rho,\phi,\theta)} = \begin{pmatrix} \sigma_{\rho\rho} & \sigma_{\rho\phi} & \sigma_{\rho\theta} \\ \sigma_{\rho\phi} & \sigma_{\phi\phi} & \sigma_{\phi\theta} \\ \sigma_{\rho\theta} & \sigma_{\phi\theta} & \sigma_{\theta\theta} \end{pmatrix} = J\,\Sigma_{(x,y,z)}\,J^T, \quad (6)$$

where J is the Jacobian of Eq. 5 and $\Sigma_{(x,y,z)}$ is defined in Eq. 1. The probability density function (PDF) for each voting cluster is given by

$$f(q) = \frac{1}{\sqrt{(2\pi)^3\left|\Sigma_{(\rho,\phi,\theta)}\right|}}\,\exp\left(-\frac{1}{2}\,\delta^T\,\Sigma^{-1}_{(\rho,\phi,\theta)}\,\delta\right), \quad (7)$$

where $\delta = q - \mu_{(\rho,\phi,\theta)}$. Here, $|\square|$ and $\square^{-1}$ are the determinant and the inverse of a matrix, respectively. When evaluated for each accumulators map cell, Eq. 7 guides the proportion of votes that each bin must receive. The voting step is performed in a flood fill style, starting from the bin associated to $\mu_{(x,y,z)}$ and using as stopping criteria two standard deviations from the mean plane parameters, *i.e.*, the Gaussian distribution is truncated in two standard deviations.

C. Peak Detection

Once the voting process is done, the local maxima address in the accumulators map correspond to the parameters of the detected planes (Fig. 2, step iii). By considering that we have a discretized parameter space, there might be votes for the same plane distributed in adjacent cells. Thus, in order to consolidate adjacent peaks and avoid detections of spurious planes, it is necessary to apply a low-pass filter [4] to smooth the accumulators map.

Our peak detection approach is based on the idea that there is consistency between the parameters associated with the peaks of votes and the parameters associated with each cluster that voted to compose that peak. Thus, the $\mu_{(\rho,\phi,\theta)}$ points are likely to be close to the peak of votes in parameter space. Thus, they may provide a good first guess to apply a hill climbing strategy, searching for the local maxima. Our peak detection procedure consists of selecting the parameters of cluster mean planes and check whether is a local maximum. If not, we move to the location of the neighbor having the higher value and repeat the process until finding a peak of votes.

D. Time Complexity Analysis

For the time complexity analysis of the proposed approach, we are going to assume as input a depth image with $d \times d$ pixels and $n = d^2$ points, with $d \geq 2$ being power of two.

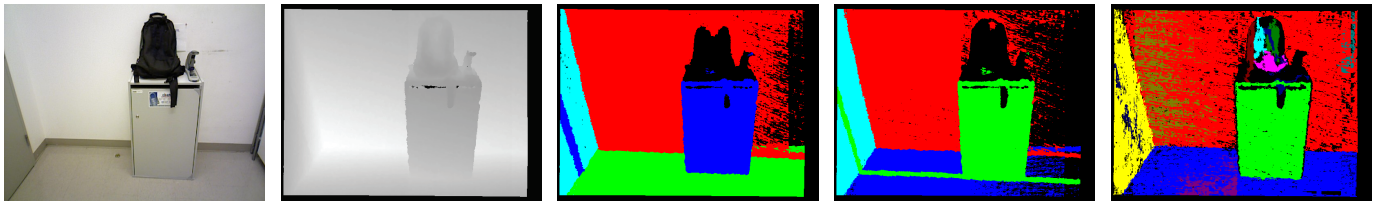
The first step of the presented approach is to clusterize the input points. At first, the SATs are precomputed and the implicit quadtree is built for storing the clusters of approximately coplanar points. The cost of building each SAT is $O(n)$ [10]. In the worst case, the quadtree would subdivide completely and the tree leafs would have size 2×2. Thus, the height of the tree would be $\log_4 n$, leading to $\sum_{l=1}^{log_4 n} 4^{l-1} = (n-1)/3$ nodes. Per node, a 3×3 covariance matrix, $\Sigma_{(x,y,z)}$, its eigen decomposition and the mean point $\mu_{(x,y,z)}$ are computed. The SATs allow the computation of the covariance matrix and the mean point in $O(1)$ time, as well as the decomposition of $\Sigma_{(x,y,z)}$. Considering all the nodes, the clustering step is performed in $O(n)$.

In the voting step, we must compute the cost for the first-order error propagation, which is $O(1)$ per cluster. Also, the cost for updating the accumulators map is $O(1)$, per cluster. Notice that the number of accumulator cells updated depends on the distribution of points in the cluster, and on the discretization step adopted for the accumulator. It is possible to assume that it is much smaller than n, which may be considered a constant value, leading to time $O(1)$ per cluster. In worst case we have $n/4$ clusters, thus the asymptotic time for the voting step is $O(n)$.

Finally, the filtering step of an accumulator cell takes $O(1)$ time and has a total cost of $O(n)$. Notice that it is only necessary to filter the bin that received votes, *i.e.*, there is no need to iterate over the entire accumulators map. By using mean planes of each cluster as first guesses, hill climbing makes a few accesses to accumulator bins per climbing (much smaller than n). Thus, the time complexity of the peak detection step is $O(n)$. The total time complexity of the D-KHT is $O(n + n + n) = O(n)$.

E. Discussion

Our approach is, just like 3-D KHT [7], a specialization of KHT [4] for a specific dimensionality and type of input. Despite using the same model of accumulators map [13] and share the same KHT pipeline, the D-KHT and 3-D KHT have fundamental differences. By making use of an octree to organize the point cloud and not counting with input regularity, the 3-D KHT has asymptotic cost $O(n \log_8 n)$. The computational cost for the voting step is the same for both techniques. To find local maxima, however, the 3-D KHT sorts the accumulator bins by the number of votes in descending order and iterate over accumulator detecting cells that are not adjacent to the already inspected ones, leading to $O(n \log n)$ cost. Thus, the overall complexity of 3-D KHT is $O(n \log n)$. The D-KHT, on the other hand, is not able to handle unorganized point clouds, since it assumes regular 2.5-D data as input.



(a) Color image (b) Input depth image (c) D-KHT (d) 3-D KHT (e) SG

Fig. 4. Frame 4161 from the *Copy Room* dataset. It compares the detection capability of the techniques in a noisy scenario (best viewed in color). From left to right: (a) color image; (b) input depth image; and (c)-(e) planes detected in (b) by D-KHT, 3-D KHT, and SG, respectively. Notice that all the techniques are able of retrieving the main planes of the scene although the D-KHT is ~ 4 times faster than the 3-D KHT and SG in this example. Notice also that the the presence of a non-planar object (the schoolbag in the scene) leads to spurious detections in SG and that the plane detection performed by D-KHT retrieves planes fits better to the points.

IV. Experiments and Results

The experiments, better described in Chapter 5 of [5], were planned to verify two aspects of the D-KHT, when compared to state-of-the-art techniques designed for unorganized point clouds (3-D KHT) and for depth images (SG).

The 3-D KHT implementation was provided by Limberger and Oliveira and the SG implementation was obtained from `OpenCV` library, which is based in Poppinga *et al* [8] and Xiao *et al.* [9] approach. All the techniques were implemented in C++ and compiled with Visual Studio 2013. The experiments were performed on an Intel Core i7-4770 64-bits CPU with 3.4 GHz and 16 Gb of RAM. It is important to note that the implementations of D-KHT and SG are totally sequential, while 3-D KHT uses `OpenMP` to parallelize the clustering step. Both D-KHT and 3-D KHT used `dlib` for handling eigendecompositions.

We used five datasets in order to evaluate the execution time, including three synthetics and two real. The *Living Room*, whose frame 294 is presented in Fig. 1 and the *Office*, are synthetic datasets provided by [6]. The third synthetic dataset, the *Occlusion Room*, is a controlled environment designed in this work to evaluate the quality of the detections. Notice that synthetic datasets, differently from real datasets, do not include systematic errors introduced by the capture device. Therefore, we also have used two real datasets: the *Copy Room*, captured by [14] with an Asus Xtion PRO LIVE camera, whose frame 4161 is presented in Fig. 4, and the *Cube*, whose single frame is presented in Fig. 3, was captured by us using a Structure Sensor. All the used datasets have depth images with 640×480 pixels. For performance evaluation of the 3-D KHT, we have converted depth images into point clouds.

The experiments show that the our approach was able to achieve ~ 119.4 fps for simpler depth images captured from real scenes (*e.g., Cube* dataset, frame 1), and from ~ 94.2 to ~ 142.6 fps for more complex real inputs (*e.g., Copy Room* dataset, frames 1063 and 4161). The 3-D KHT, in contrast, was not able to perform real-time detection in most of the datasets, except for frame 1063 of the *Copy Room* dataset. For other depth images of real scenes, the detection was performed in a range from ~ 9.5 to ~ 26.9 fps. SG could not achieve real-time performance, unless the depth map and the normal map were given as input.

The D-KHT clustering step, considering the SATs application, runs from ~ 3 to ~ 9 times faster than the clustering strategy adopted in the 3-D KHT. For synthetic scenes like *Living Room* and *Office*, considering that the input depth images has no missing information, the voting procedure dominates the time of both techniques. For real datasets, where the points are not perfectly aligned over the planes and there are areas with missing depth information, the clustering step dominates the time of the D-KHT (except for the *Cube* dataset), while the local maxima detection is the slower step of the 3-D KHT (which does not happens for frame 1063 of the *Copy Room* dataset). Finally, the hill climbing strategy has produced a noticeable difference in local maxima detection performance in relation to 3-D KHT. The hill climbing strategy leads to a peak detection ~ 22 times faster than the 3-D KHT in this step (*e.g., Copy Room* dataset, frame 1791). For SG, the normal map computing had the longest execution time for all the datasets but since devices like Structure Sensor and Project Tango provide the normal map with the depth images, we can consider, in this cases, only the detection time. The detection performance achieved was from ~ 39.5 fps to ~ 51.2 fps for synthetic scenes and ~ 34 fps to ~ 47.8 fps for real scenes. The experiments has demonstrated, however, that the D-KHT is faster than the detection step of the SG.

For the quality measurement of the detection, we have created and used the *Occlusion Room* dataset, which consists of spread geometric shapes with a known location, size, and orientation. For each plane, the parameters ϕ, θ and ρ are known. The experiment is based on checking how close the planes detected by D-KHT, 3-D KHT, and SG are to the previously known plane parameters. In order to evaluate the quality of the detection, we have used two metrics: (i) the *cosine similarity* between the unitary normal vectors of the planes, where values close to *one* indicate similar parameters; and (ii) the absolute difference of the *Euclidean distances* between the detected planes and their reference planes to the origin of the space, where values close to *zero* indicate better fit. The means and standard deviations are summarized in Table I. Notice that the plane parameters retrieved by the D-KHT are closer to the ground truth parameters than other techniques. The differences are negligible but SG has a lack of resilience to noise.

TABLE I
QUALITY OF DETECTIONS IN DATASET *Occlusion Room*.

	Cosine Similarity		Euclidean Distance	
	Mean	Standard Deviation	Mean	Standard Deviation
D-KHT	0,99736	0,00343	13,63	12,58
3-D KHT	0,99151	0,01903	14,53	13,14
SG	0,99700	0,00479	13,99	11,81

TABLE II
AMOUNT OF DETECTED PLANES IN DATASET *Occlusion Room*.

	D-KHT	3-D KHT	SG
Single detection of existing planes	14	10	10
Multiple detection of existing planes	0	0	8
Missing planes	3	7	4
Detection of spurious planes	0	2	3

Table II summarizes the amount of planes detected by each technique in the *Occlusion Room* dataset. By observing this table, one is able to notice the multiple detections of the same plane, a common issue in SG, avoided by HT-like approaches. The main cause of this problem is occlusion and discontinuity.

V. PUBLICATIONS

As a product of this work, we have submitted a paper to the *Pattern Recognition Letters* journal, and have received a positive feedback, considering that the paper has passed the first round of revisions. The draft of the paper is attached to this submission. The deadline for submitting the revised version of the manuscript is August 17th.

VI. CONCLUSION

In this work, we presented a real-time approach to plane detection in depth images. To ensure the low computational cost of the technique, we took advantage of a few restrictions from the capture model and from the regular structure of the depth images.

The effectiveness of the method was demonstrated with its comparison to state-of-the-art techniques when applied to datasets comprised of both synthetic and real images. Besides that, the analyzed datasets also had non-planar surfaces to evaluate algorithms resilience to detecting spurious planes, a common issue in this kind of technique. The D-KHT proved to be effective for depth images, being executed from 3 to 5 times faster than state-of-the-art.

It is important to highlight that this work comes from the need of a real-time solution for plane detection in scenes. Such need comes from a larger project developed in *Visgraf* and *Prograf* research groups, which aims to approximate the geometry of indoor environments by planes by using mobile devices, as discussed in [5]. Thus, it is possible to notice that this work already has practical applications, and can even be implemented in mobile devices, given its performance.

ACKNOWLEDGMENTS

We thank Oliveira, Limberger, and Breda for kindly providing the reference implementation of the 3-D KHT, and Souza for modeling the *Occlusion Room*. This work was sponsored by CNPq-Brazil grants 456.016/2014-7, 308.316/2014-2, and FAPERJ grants E-26/110.092/2014, E-26/202.832/2015.

REFERENCES

[1] Occipital, "Structure sensor," 2015. [Online]. Available: http://structure.io

[2] Google, "Project tango," 2015. [Online]. Available: https://www.google.com/atap/project-tango/

[3] P. V. C. Hough, "Machine analysis of bubble chamber pictures," in *Proc. of Int. Conf. on High Energy Accelerators and Instrum.*, 1959, pp. 554–558.

[4] L. A. F. Fernandes and M. M. Oliveira, "Real-time line detection through an improved Hough transform voting scheme," *Pattern Recognit.*, vol. 41, no. 1, pp. 299–314, 2008.

[5] E. Vera, "D-KHT: Detecção em tempo real de plano em imagens de profundidade," Master's thesis, Insituto de Computação, Universidade Federal Fluminense, Niterói, Rio de Janeiro, Brasil, 2016. [Online]. Available: http://www2.ic.uff.br/PosGraduacao/Dissertacoes/731.pdf

[6] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *Proc. of CVPR*, 2015, pp. 5556–5565.

[7] F. A. Limberger and M. M. Oliveira, "Real-time detection of planar regions in unorganized point clouds," *Pattern Recognit.*, vol. 48, no. 6, pp. 2043–2053, 2015.

[8] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak, "Fast plane detection and polygonalization in noisy 3D range images," in *Proc. of IROS*, 2008, pp. 3378–3383.

[9] J. Xiao, J. Zhang, J. Zhang, H. Zhang, and H. P. Hildre, "Fast plane detection for SLAM from noisy range images in both structured and unstructured environments," in *Proc. of ICMA*, 2011, pp. 1768–1773.

[10] F. C. Crow, "Summed-area tables for texture mapping," *SIGGRAPH Comput. Graph.*, vol. 18, no. 3, pp. 207–212, 1984.

[11] H. Samet, "The quadtree and related hierarchical data structures," *ACM Computing Surveys*, vol. 16, no. 2, pp. 187–260, 1984.

[12] L. Parratt, "Probability and experimental errors in science," *J. Frankl. Inst.-Eng. Appl. Math.*, vol. 272, no. 6, pp. 527–528, 1961.

[13] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter, "The 3D Hough transform for plane detection in point clouds: A review and a new accumulator design," *3D Research*, vol. 2, no. 2, pp. 1–13, 2011.

[14] Q.-Y. Zhou and V. Koltun, "Dense scene reconstruction with points of interest," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 112:1–112:8, 2013.