Robust Room-Structure estimation in Manhattan-like Environments from dense $2\frac{1}{2}$D range data

Sven Olufs and Markus Vincze

Abstract— **In this paper we propose a novel approach for the robust estimation of room structure using Manhattan world assumption i.e. the frequently observed dominance of three mutually orthogonal vanishing directions in man-made environments. First, separate histograms are generated for every major axis, i.e. X, Y and Z, on stereo data with an arbitrary roll, pitch and yaw rotation. These histograms are maintained in the fashion of quadtrees. Using the traditional Markov particle filters and minimal entropy as metric on the histograms, we are able to estimate the camera orientation with respect to orthogonal structure. Once the orientation is estimated we extract hypothesis of the room structure by exploiting 2D histograms, i.e. X/Y, Z/Y, Z/X, using mean shift clustering techniques. Finally, the hypotheses are evaluated with the real data and false hypothesis are pruned. We also show the robustness of our approach with respect to noise in real world data.**

I. INTRODUCTION

The estimation of room-structure, e.g. corridors, door or walls, is a vital task for mapping or navigation. With domestic robotics we face the problem of clutter and visually weak by structured environments. Here the use of 2D sensors like laser range scanners is limited, e.g. if the environment is only partially known [1]. In the last years the use of stereo cameras for perception has become quite popular after the pioneering work of Jim Little et al. [2] in 2000. The suitability of stereo vision for indoor environments has been shown in various works, e.g. VisualSlam [3], safe navigation [4] or obstacle avoidance [5]. Another trend is the use of the time of-flight-cameras, e.g. the CSEM Swissranger 3000 [6] for this kind of applications.

The challenge with data from $2\frac{1}{2}$D depth data is to cope with noise and uncertainty due to the nature of the sensors. The certainty of depth data from time-of-flight cameras depends on the material (e.g. not shiny or light absorbing in the infrared spectrum) of the environments, which usually leads to noise in depth estimation. Depth data with stereo vision is estimated by matching corresponding pixels (or small patches) in the image pair. It is assumed that the true corresponding pixels (or patches) are distinguishable from those surrounding structure. Within the domestic robotics domain the environments can be very ”non-discriminative”, e.g. single-coloured walls or furniture, so it can result in a few certain and many uncertain estimates. Certain estimates can result from the boundaries of objects. Here the usage of parametric fitting methods, e.g. the common RANSAC based plane estimation [7], is limited due to the high probability of false positives. Such false positives can be plausible to fitting method because depth estimates are not equally distributed. Another issue is that the sensor depth resolution does not scale linearly in most stereo vision systems. Due to the system depth estimate of far objects is always less certain than of close objects.

(a) Financial District, Manhattan / NY (b) Broadway Street, Manhattan / NY

Fig. 1. Principle of the Manhattan like environments. Figure 1(a): The structure is aligned to the three major axis, i.e. the walls are aligned parallel. Figure 1(b) shows two Manhattan world configurations: The dominant one is Manhattan itself, the other one is the Broadway Street which shows a partial Manhattan-like structure.

Many approaches for room-structure estimation use the concept of occupancy grids [8] or extensions to 3D, e.g. [9]: The grid contains information on a quite primitive level if a grid cell is a wall or ground. At this level there is no information if certain parts of grid cells with the label ”wall” are aligned to other ”walls” or if the ground is parallel to other structures, e.g. a table top. This kind of constrains is referred to in the computer vision literature as the so-called *Manhattan world* assumption. I.e. the frequently observed dominance of three mutually orthogonal vanishing directions in man-made environments [10], see figure 1. Many indoor environments can be considered as Manhattan-like or quasi Manhattan-like, e.g. a couch can be aligned to a wall or the walls within a corridor are usually parallel to each other and orthogonal to the ground.

In this paper we propose a novel approach, for the robust estimation of room-structure using Manhattan world assumption. The approach extracts a structure which is parallel to one of the major axis i.e. X, Y and Z. First the step of our approach finds the initial camera orientation, namely roll,

Sven Olufs and Markus Vincze are with the Vienna University of Technology, Automation and Control Institute, Gusshausstrasse 25-29 / E376, A-1040 Vienna, Austria

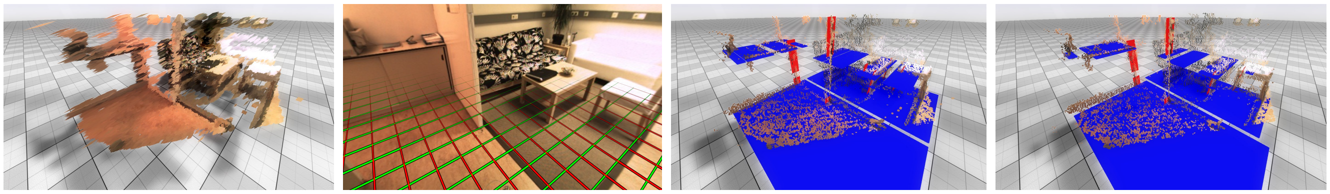

(a) 3D Stereo Vision (b) Estimated Camera Orientation (c) Plane hypotheses (d) Final Output (after pruning)

Fig. 2. Overview of our approach for stereo vision data. Please note that the back of the couch is not extracted on purpose since its tilted.

pitch and yaw using the principle of minimum entropy in histograms. Next we, extract hypotheses of the room structure by exploiting 2D histograms i.e. X/Y, Z/Y, Z/X using mean shift clustering techniques. Finally the hypotheses are evaluated with the real data and false hypotheses are pruned. The paper is organized as follows: After discussing the state of the art, we describe in section III the proposed approach. Next we present experimental results, followed by applications of the approach in robotics. Finally, we give a conclusion in section V.

II. RELATED WORK

The literature proposes various techniques for estimating room structure. Triebel et al [11] use a hieratical expectation maximization method to extract planes from 3D laser range scans. Other methods use an octree over segmentation [9], RANSAC [12] or rely on split and merge plane strategies [13]. A common approach is to first estimate the dominant ground plane in the image in order to obtain the pitch angle by exploiting the kinematics of the robot. Murarka et al [14] use a parametric plane fit on segmented disparity patches to obtain the camera pose. The segmentation is based on colour and local homogeneity in the image and on exhausting graph search. The plane fit is merged to a plane hypothesis using graph cuts and energy minimization [15]. Graph cut and segmentation are both computationally expensive with $O(n^2)$ for n pixel. Yu et al. [16] use the normal vectors of the disparity map to estimate planes by grouping similar vector directions in the neighbourhood to planes with its normal vector (mainly) pointing in Y-direction. The largest plane with a dedication to the Y axis is used as an estimate. Yu et al. assume that the ground is always the largest object in the image. Another approach is proposed by Burschka et al. [5] by learning the parameters of the ground plane for re-detection. The approach is quite robust, but assumes a ground parallel camera with very little roll rotations while it is able to handle changing pitch.

The use of the *Manhattan-World* assumption is quite popular in the computer vision literature, for instance, in the use of multi view-reconstruction [17], [18], [19]. Gallup et al. [17] use *Manhattan-World* assumption as prior for plane sweeping i.e. using only orthogonal planes. Furukawa et al. [19] use a similar approach for reconstructing piecewise planar patches and Markov random field formulation for exact planes. Sinha et al. [18] use a similar method, but with a less strict model.

III. OUR APPROACH

The main idea of our approach is to use histograms to extract room structure hypotheses rather than using the depth data as voxel. One advantage of using histograms is that it is relatively easy to estimate the Manhattan-like structure within the data using the principle of minimum Shannon entropy. We estimate the relative camera orientation to the Manhattan-like structure (roll, pitch, and yaw) so that the dominant structure is aligned to all three axes (X,Y and Z). One disadvantage is that we lose spatial information about the voxels i.e. post processing is needed to generate hypotheses on the room structure on all three axis i.e. planes aligned to the X, Y and Z axis. The hypotheses are finally evaluated using the $2\frac{1}{2}$D depth data in the fashion of plane sweeping [17].

A. Pre-processing

The main issue with $2\frac{1}{2}$D depth data from stereo vision or time of flight cameras is uncertainty. One way to cope with this is to use an ellipsoidal representation for uncertainty of the individual voxels. Depending on the sensor we can define a metric to estimate the uncertainty of the individual voxel. For the sake of simplification we set the volume of the ellipsoid always to 1. We use the 3D coordinates of the individual voxels $v^{i=1..n}$ as centre of the ellipsoid. The major axis the orientated to the focal point of the vision system since we use only $2\frac{1}{2}$D data. The length a^i of the major axis depends on the used sensor (see below). Finally, we use a fixed 1:1 ratio for minor and vertical axis, the

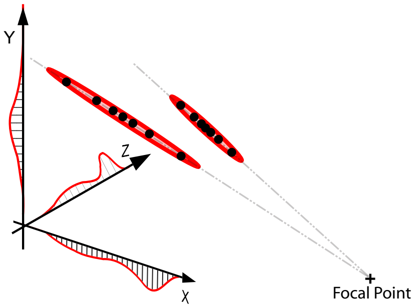


Fig. 3. The uncertainty of the data, shown as red ellipsoids, is approximated using additional voxels.

length of these axes can be easily obtained due to the kown volume and major axis length of the ellipsoid.

The mapping of the ellipsoids to the histograms is done by approximating the ellipsoids as additional voxels, see fig. 3: Here is the main idea to approximate the density of the ellipsoid with additional set of voxels in the fashion of particle filters. Since the minor and vertical axes of the ellipsoid are fixed we only approximate the major axis with voxels. This approximation is sufficient as histograms for data processing are used. Additional voxels are drawn as follows: Let $m \in \mathbb{N}$ be the level of interpolated voxels $v' = \{v_{-m}, .., v_m\}$ for one individual voxel v and let $v_j = \{-m, .., m\}$. The new interpolated voxel v' is given using the Gaussian normal distribution as

$$v' = v + a^i c_g \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{v_w^2}{2\sigma^2}}$$

with c_g as normalizing constant and $v_w = \frac{v_j}{m+1}$. The term v_w prevents that new voxels are drawn at the borders with small m, e.g. $m < 2$. For the sake of simplicity we assume the same $\sigma = 1$ for all ellipsoids. A similar approach has been used by Sinha et al. [18].

1) Stereo Vision: The length a^i of the major axis is calculated as follows: First we back-project the v^i voxel to a disparity value $d^i \in \mathbb{N}$ of the stereo vision system. Next we calculate the offset in the disparity space with $d^i = d^i - (1 + c^i f)$ where c^i is the confidence of the voxel from the stereo matching and f is a normalizing constant. Here we assume that a small disparity value represent a farther distance than a large one. Finally, we back-project the new disparity value d^i into the Cartesian space and let be a^i the Euclidian distance of v^i and the shifted back projected disparity value. We want to emphasize that the stereo vision does not scale linear in the conversion from disparity values and 3D voxels.

2) Time-Of-Flight Cameras: In the case of the ToF sensor the length a^i of the major axis is calculated using the corresponding amplitude of the individual depth data. The length a^i is approximated using a Lorentz function [7] using the amplitude v_a^i as input

$$a^i = y_0 + \frac{2A}{\pi} \frac{w}{\pi} \frac{w}{4(v_a^i - xc)^2 + w^2}$$

with A as area[1], xc as centre[2] , w as weight[3] and y_0 as offset[4] of the Lorentz function. The parameters have been estimated through a nonlinear Marquardt-Levenberg optimization and representative ground truth data.

B. Minimum Entropy in Histograms

The main idea of our approach is to estimate the camera orientation using the principle of minimum entropy of histograms. First three independent 1D histograms for X, Y and Z are built from an arbitrary configuration (α,β,γ) i.e. the data is rotated with γ yaw, then β pitch and α roll. Next the Shannon entropy $H(X)$ of all three normalized histograms X_x, X_y, X_z is calculated with

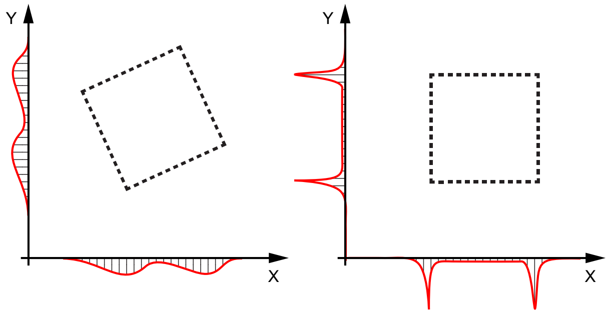


Fig. 4. Basic Idea of estimation using minimum entropy: Both figures contain the same data, but with different rotation. The entropy of the histograms in the right figure is significantly smaller than the entropy on the left histograms.

$$H(X) = -\sum_{i=1}^{k} p(x_i) \log_{10} p(x_i)$$

where k is the number of bins of the histogram and $p(x_i)$ is the value in the histogram at bin i. In the case of $p_i = 0$ for some i, the value of the corresponding summand $0 \log_{10} 0$ is taken to be 0, which is consistent with the limit $\lim_{p \to 0+} p \log_{10} p = 0$.

The Manhattan configuration estimate is obtained with

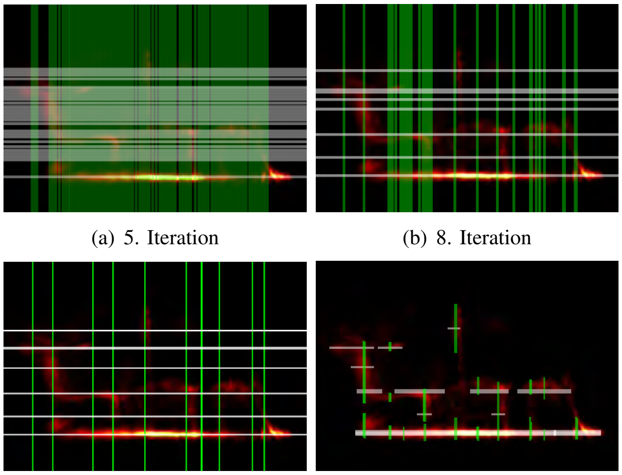
$$\operatorname{argmin}(H(X_x) + H(X_y) + H(X_z))$$

i.e. the configuration with the lowest entropy. Similar approaches have been used by Gallup et al. [17] for multi view reconstruction and Saez et al. [20] for vision based SLAM.

In practise we use a particle filter to estimate and track the camera orientation (configuration) using 50 particles in the (α,β,γ) state space instead of using non-linear optimizers like Levenberg−Marquardt algorithm [7] which is typically used for *argmin* in the computer vision literature. There are two reasons we do not use non-linear optimizers: First, with particle filters we can easily incorporate the motion of the robot and improve the robustness of the tracking. The second reason is that the output of the optimizers depends on a ”good’ initialisation and can get stuck in local minima while we initialise the particles in the Monte Carlo fashion. In this case we favour robustness instead of accuracy.

Figure 4 depicts the overall concept: If there are parts in the image which are orthogonal to one of the X,Y,Z axis, i.e. Manhattan world assumption, then it is possible to find a configuration using minimum entropy. It is not necessary that all parts in the image are Manhattan-like or directed to its main axis as long there is some structure orientated to

[1] $3.268694711152766 \ 10^4$
[2] $1.711211949550388 \ 10^1$
[3] $4.224211770198602 \ 10^{-1}$
[4] $-3.503781396582853 \ 10^{-3}$



(a) 5. Iteration

(b) 8. Iteration

(c) 30. Iteration

(d) Edge grouping with hysteresis

Fig. 5. Constrained mean shift clustering on the X/Y histogram of Figure 2(a)

the major axis. Please note that fig. 4 shows the concept for the 2D case for the sake of simplicity. We also want to emphasize that our approach can also deal with occlusions due to the fact we build histograms direct on voxel in contrast to many monocular vision based approaches [17], [18], [19].

C. Generating Plane Hypotheses

Once the camera orientation is known, we can generate plane hypotheses from the upsampled voxel data. This is done in three steps: First, generate individual 2D histograms on all possible X,Y,and Z combinations i.e. X/Y, Z/X and Z/Y. In the next step, extract line segments in the 2D histograms and group them to planes in the last step.

The generation of the 2D histograms is straight forward: First, an inverse rotation is applied to the upsampled voxels using the known camera orientation. This aligns the voxels to one of the three major axis. Usually not all parts of an image are aligned to the major or dominant axes. Here we assume that the majority of data is aligned to the dominant axes. It is possible to recover all local Manhattan configurations, but this is not feasible due to the problem beeing NP hard. For the histograms itself we use a resolution of 5cm per bin. Please note that the resolution of stereo cameras does not scale linearly, but we are able to overcome this, by using voxel upsampling .

As we use Manhattan-like environments the line extraction can be constrained to vertical and horizontal lines only. The extraction itself is done on all in two steps: First, we group segments in the histogram together using the mean shift algorithm [21] and edge grouping with hysteresis in the fashion of the canny edge detector [22].
Mean shift itself is a procedure for locating the maxima of a density function given discrete data sampled from that function. In our case we use it for detecting the modes of this density. This is an iterative method, and we start with an initial estimate x. Let a kernel function $K(x_i - x)$ be given. This function determines the weight of nearby points for re-estimation of the mean. We use the Epanechnikov kernel

$$K(x) = \begin{cases} 1 - \|x\|^2 & if\|x\| \le 1 \\ 0 & if\|x\| > 1 \end{cases}$$

on the distance to the current estimate,

$$K(x_i - x) = e^{c\|x_i - x\|}$$

The weighted mean of the density in the window determined by K is

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) x_i}{\sum_{x_i \in N(x)} K(x_i - x)}$$

where $N(x)$ is the neighbourhood of x, a set of points for which $K(x) \neq 0$. The mean-shift algorithm now sets $x \leftarrow m(x)$, and repeats the estimation until $m(x)$ converges to x or a the maximum of iterations is reached.

Instead of applying the mean shift on the entire 2D histogram, we first reduce the 2D histograms to two 1D histograms and apply the mean shift separately (due to the Manhattan world assumption), see figure 5. We use a parameterized Epanechnikov kernel with a size that represents 7.5cm in the real world. In the next step we use the output of the mean shift clusters as input for the line boundary detection using to "other" 1D histogram: For instance we use a X/Y histogram. We apply the mean shift on the X 1D histogram to determine the height of the line and use the Y to estimate the boundaries of the line or lines. The boundaries are estimated in the hysteresis fashion e.g. as used in the canny edge detector. The algorithm assigns first

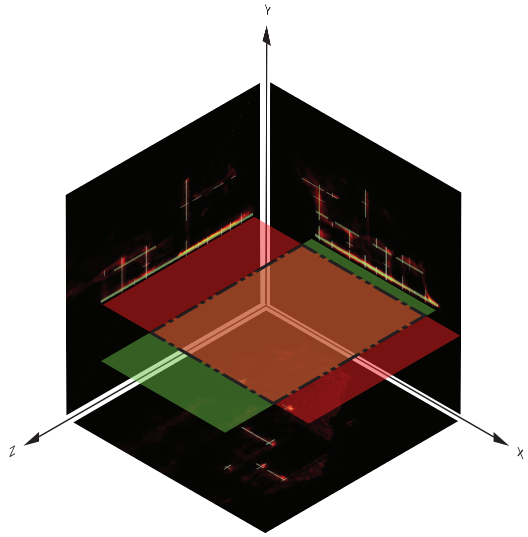


Fig. 6. Principle of the plane hypothesis generation of Figure 2(a). The line segments of two orthogonal aligned histograms are used to generate planes. The intersected area is the actual output. Note that it is possible that more than one plane hypothesis is generate per one line.

labels ”no edge” (0) ”maybe edge” (1) and ”edge” (2) to each pixel within one line. Instead of using only the pixels within one mean shift cluster we use the maximum value within the variance of the mean shift cluster, e.g. in a horizontal cluster ”line” we also consider pixels that are ”above” and ”below” the line. The variance is calculated using simple backtracking, see [21] for details. Let T_m and $T_h, T_h < T_m$ be two thresholds. T_m denotes the ”edge” threshold while T_h is the hysteresis threshold. Let q_i be pixel value from the cluster at position i and

$$C(p_i) = \begin{cases} 2 & if||p_i|| \geq T_m \\ 1 & if||p_i|| \geq T_h \\ 0 & if||p_i|| < T_h \end{cases}$$

be a function that assigns labels to the pixels p. Next the algorithm assigns the label ”edge” to labels with ”maybe edge” if an ”edge” pixel is nearby within the recursive function

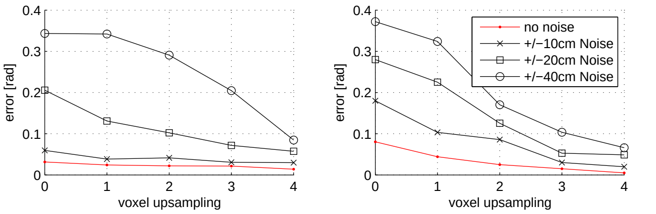
$$C'(p_i) = \begin{cases} 2 & ifC(p_i) = 2 \\ 2 & ifC(p_i) = 1 \wedge (C'(p_{i-1}) = 2 \vee C'(p_{i+1}) = 2) \\ 0 & ifC(p_i) = 0 \end{cases}$$

Finally we group all ”edge” pixels to line segments using simple run length encoding, see figure 5(d).

Now we group the line segments to plane hypothesis. Since each line segment is aligned with one major axis (X, Y, or Z), we can generate plane hypotheses by projecting planes to the normal axes of the parent 2D histograms of two axes. For example ”horizontal ground planes” can be generated using the X/Y and Z/Y histograms and their corresponding line segments (for X and Y line segments with the same height from the ground). All ”X” line segments of X/Y histogram are projected orthogonally to the Z axis. Next we project the ”Z” line segments of the Z/Y histogram and build planes the same way. The intersected area of two planes is used as plane hypothesis in the fashion of plane sweeping methods [17]. The generation of the X and Y planes is done the same way, figure 6 depicts the overall concept.

D. Pruning Plane Hypothesis

Finally, we use a pruning strategy to remove plane hypotheses with no or little support of the upsampled voxels. First, we sort the plane hypotheses f according to the joint probability $p(f) = p(l_n, l_m)$ of the line pair l_n, l_m and the size (largest first). The individual $p(l_n)$ and $p(l_m)$ result from the mean shift clustering and are normalized weights, see [21] for details. Next all plane hypotheses are evaluated by reprojecting them back into the 3D state space, counting the number of inliers per plane in RANSAC fashion. Planes with no support $count < 0.1\%$ are then removed from the set. This step removes 90% of all false planes. Another $approx 8.5\%$ can be removed in the fashion of plane sweeping: If a plane with a low probability occludes the visibility of a higher one, it is removed from the set. The visibility check is ego centred at the focal point.



(a) Stereo Vision (AIT) (b) Swissranger (SR-3000)

Fig. 7. Average angle error of the tours with different levels of artificial Gaussian noise.

IV. EXPERIMENTAL RESULTS

For our experiments we use a non-holonomic mobile robot manufactured by Bluebotics with an additional SICK LMS 200 laser range finder mounted to its front. We use the AIT Stereo Camera, with two b/w HDR sensors, a colour sensor in the centre and approx. 100 degrees field of view. Only the two left and right HDR sensors are used for dense stereo vision, the output image is projected onto the centre camera. The sensor is oriented approx 35 degrees downwards woth respect to the robot's driving direction and mounted at a height of a 100 cm over the ground. We use a GPU implementation of the CENSUS AIT Stereo Engine [23] for dense stereo-data calculation at a resolution of 720x480 and using 80 disparities. The GPU implementation of the engine enables us to work up to 120fps on a NVIDA Geforce GTX 280 and 40fps on a MacBook Pro with an Nvidia Geforce 9600.

We choose a typical home environment (see fig. 2) for data acquisition using the stereo camera system and laser data. Ground truth for roll and pitch is obtained using Nicolas Guilbert's [24] structure from motion toolbox[5], yaw is obtained from the robot pose of the laser-based self-localisation. While structure from motion approaches determine only relative motion we use an additional IMU for roll and pitch initialisation. The data of all sensors is recorded at 25 frames per seconds. All stereo data is calculated off-line from the previously recorded images. We recorded representative six tours through our lab with a total length of approximately 250 meters. Three tours have a Manhattan like environment while the other ones represent a quasi Manhattan like environment. The robot moves with an average travelling speed of $0.65\frac{m}{s}$.
Figure 7 shows the average angle error for all tours using our approach with different amounts of Gaussian noise. One can see that the upsampling of voxels has an impact on the noise level. The Swissranger shows faster convergence due to the data is more dense than stereo vision.

Finally, we consider the runtime of our approach: Table I depicts the runtime for various configurations. Our code is run on 2.4 GHz QuadCore PC, while the code is not

[5] http://www.maths.lth.se/matematiklth/personal/nicolas/octave-vision.html

TABLE I

Average Runtime of our approach using histograms with 1024 bins. The time measurements are normalized for 100000 voxels and $m = 3$ upsampling

voxel upsampling	2ms
camera orientation estimation (50 particles)	38ms
2D Histogram generation	1ms
Mean shift clustering	5ms
Edge Grouping with hysteresis	1ms
Plane hypothesis generation	2ms
Plane hypothesis pruning	21ms
Sum	72ms

optimized and uses only one CPU (except for the particle filtering). The extension of the code to multithreading, i.e. using multiple CPUs, is straightforward. One can see that the bottleneck of our approach is the calculation of the entropy due to the usage of histograms and particle filters. Using smaller histograms will result in a lower constant runtime, but will also influence the accuracy negatively.

V. CONCULSION

In this paper we presented a new robust method for estimating planes in $2\frac{1}{2}$ depth data in a Manhattan like environment. Once the orientation is estimated we calculate the dedication of every voxel to the 3 major axes and we can extract planes using histogram voting. We also showed that the method is robust to noise using an upsampling technique, based on the a-priori known uncertainty of the voxels. Our methods works with dense stereo vision and time-of-flight cameras (SR-3000), but can also be applied to 3D laser scanners or ladar sensors.

At this point we want to emphasize that our method is only appliable to $2\frac{1}{2}$D depth data and can *not* be applied to dense 3D data: Even if many false hypothesis are generated, they can easily be pruned due to missing support of voxel data i.e. only one sensor reading per $2\frac{1}{2}$ point. Our next steps will be to aim for an extended model to extract planes from a single view based on graph cuts [15].

VI. ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community's Sixth Framework Programme (FP6/2003-2006) under grant agreement no FP6-2006-IST-6-045350and FP7-ICT-2009-4-248623.

References

[1] Sven Olufs and Markus Vincze. An efficient area-based observation model for monte-carlo robot localization. In *International Conference on Intelligent Robots and Systems IROS*, St. Louis, USA, 2009.

[2] D. Murray and J. J. Little. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 2(8):161–171, 2000.

[3] Robert Sim and James J. Little. Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS), Beijing*, 2006.

[4] A. Murarka and B. Kuipers. A stereo vision based mapping algorithm for detecting inclines, drop-offs, and obstacles for safe local navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-09).*, 2009.

[5] D. Burschka, Stephen Lee, and G. Hager. Stereo-based obstacle avoidance in indoor environments with active sensor re-calibration. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[6] Stefan May, Bjoen Werner, Hartmut Surmann, and Kai Pervoez. 3d time-of-flight cameras for mobile robotics. In *International Conference on Intelligent Robots and Systems IROS 2006*, Beijing, China, 2006.

[7] R. I. Hartley A. and Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[8] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge MA, first edition, 2005.

[9] Radu Bogdan Rusu, Aravind Sundaresan, Benoit Morisset, Kris Hauser, Motilal Agrawal, Jean-Claude Latombe, and Michael Beetz. Leaving flatland: Efficient real-time three-dimensional perception and motion planning. *Journal of Field Robotics, Special Issue: Three-Dimensional Mapping, Part 1*, 26(10):841–862, 2009.

[10] James M Coughlan and A. L. Yuille. Manhattan world: orientation and outlier detection by bayesian inference. *Neural Comput.*, 15(5):1063–1088, 2003.

[11] Rudolph Triebel, Wolfram Burgard, and Frank Dellaert. Using hierarchical em to extract planes from 3d range scans. In *In Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA*, 2005.

[12] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Andreas Holzbach, and Michael Beetz. Model-based and Learned Semantic Object Labeling in 3D Point Cloud Maps of Kitchen Environments. In *Proceedings of the 22nd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[13] Narunas Vaskevicius, Andreas Birk, Kaustubh Pathak, and Jann Poppinga. Fast detection of polygons in 3d point clouds from noise-prone range sensors. In *International Workshop on Safety, Security, and Rescue Robotics (SSRR)*, 2007.

[14] Aniket Murarka, Mohan Sridharan, and Benjamin Kuipers. Detecting obstacles and drop-offs using stereo and motion cues for safe local motion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-08).*, 2008.

[15] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:2001, 1999.

[16] Qian Yu, H. Araujo, and Hong Wang. Stereo-vision based real time obstacle detection for urban environments. In *ICAR 2003âĂŤ11th Int. Conf. on Advanced Robotics*, 2003.

[17] David Gallup, Jan-Michael Frahm, Philippos Mordohai, Qingxiong Yang, and Marc Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 2007.

[18] Sudipta N. Sinha, Drew Steedly, and Richard Szeliski. Piecewise planar stereo for image-based rendering. In *Twelfth IEEE International Conference on Computer Vision (ICCV 2009)*, 2009.

[19] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Manhattan-world stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 2009.

[20] J.M. Saez and F. Escolano. Entropy minimization slam using stereo vision. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[21] Yizoung Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(8), 1995.

[22] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679–714, 1986.

[23] Martin Humenberger, Christian Zinner, and Wilfried Kubinger. Performance evaluation of a census-based stereo matching algorithm on embedded and multi-core hardware. In *Proccedings of the International Symposium on Image and Signal Processing and Analysis*, volume 6, 2009.

[24] Nicolas Guilbert, Fredrik Kahl, Karl Astrom, Magnus Oskarsson, Martin Johansson, and Anders Heyden. Constraint enforcement in structure and motion applied to closing an open sequence. In *Asian Conference on Computer Vision*, 2004.