Semantic Segmentation in Manhattan-like Environments from 2.5D data

Sven Olufs[1] and Markus Vincze[1]

[1] Vienna University of Technology, Automation and Control Institute,
Gusshausstrasse 25-29 / E376, A-1040 Vienna, Austria
(olufs@ieee.org, vincze@acin.tuwien.ac.at)

Abstract: In this paper we propose a novel approach for the robust segmentation of room structure using Manhattan world assumption. First, we estimate the Manhattan-like structure by using an MSAC variant that estimates such a Manhattan system directly from the data. Once the orientation is estimated we extract hypotheses of the room structure by exploiting 2D histograms using mean shift clustering techniques as rough estimate for a pre-segmentation of voxels i.e. their membership to planes of a certain position and orientation. Additionally we use the concept of vanishing points to extract 2D cues from the 2.5D data to improve the segmentation. We apply superpixel over segmentation on the colour input to achieve a dense segmentation. The over segmentation and pre-segmented voxels are combined using graph-cuts for a not a-priori known number of final plane segments with a label minimizing graph cut variant proposed by Delong et al. with polynomial runtime.

Keywords: Segmentation, Manhattan-System, Computer Vision

1 INTRODUCTION

The estimation of semantic room structure, e.g. corridors, doorways or walls, is a vital task for mapping or navigation. With domestic robotics we face the problem of clutter and visually weakly structured environments. The use of 2.5D sensors has become quite popular in the last decade, for instance the use of tilting 3D laser scanners or the Swissranger SR-3000. With the recent release of Microsoft's Kinect structured light sensor, the popularity of 2.5D sensors gained a boost. The Kinect sensor is suitable for the task for two reasons: The sensors are cheaper than laser scanners and they offer an 2D colour image, which can be used for more sophisticated feature extraction.

The challenge with data from 2.5D data is to cope with noise and uncertainty due to the nature of the sensors. For instance, the quality of 2.5D data from the Kinect depends on the reflection properties of the observed surface or the angle of incidence. Since the sensor uses structured light in the infrared spectrum, the sensor is sensitive to sunlight. Within the domestic robotics domain, the environments can be single-coloured walls or furniture without texture, so it can result in few certain and many uncertain estimates. Another issue is that the sensor's depth resolution does not scale linearly with the Kinec.

Many approaches for room structure estimation use the concept of occupancy grids [3] or extensions to 3D, e.g. [4]: The grid contains information on a primitive level if a grid cell corresponds to oder belongs to 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 constraints is referred to in the computer vision literature as the so-called *Manhattan world* assumption; The frequently observed dominance of three mutually orthogonal vanishing directions in man-made environments [5, 6, 7, 8]. Many indoor environments can be considered as Manhattan-like since most walls of a room are aligned orthogonally to the ground or quasi Manhattan-like if the walls are not aligned orthogonally to each other. In many cases, furniture is also aligned Manhattan-like to its environment, e.g. a couch or cupboard can be aligned to a wall. Here we emphasize that it is not necessary that the furniture is aligned to all three major axes i.e. even if a table is not aligned to a wall its table surface is usually parallel to the ground.

The novelty of the paper is the use of 2D and 3D features in a unified framework at almost the same runtime as the previous approach in polynomial runtime unlike than common NP hard solutions. The extraction of 2D cues is done in linear runtime and improves the overall precision of the segmentation and also improves the robustness to false matches.

2 OUR APPROACH

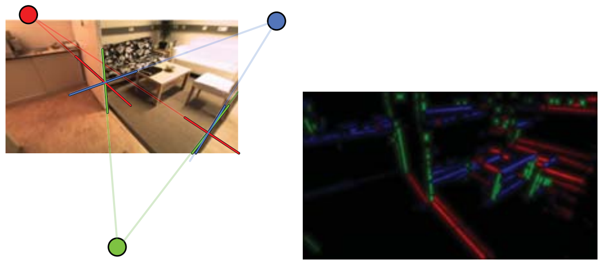
The main idea of our approach is to use histograms to extract room structure hypotheses for MRF segmentation rather than using the depth data as voxels. One advantage of using histograms is that it is relatively easy to estimate the Manhattan-like structure within the data if the camera origination is known. 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 axes i.e. planes aligned to the X, Y and Z axes. The hypotheses are finally evaluated using the 2.5D data in the fashion of RANSAC and pre-segmentation using over segmentation techniques in the source 2D image. In an additional step we extract 2D cues from the 2.5D data using the concept of vanishing points. We use the 2D cues in two ways: Depending on the structure in the image it can be sometime easier to describe which part of an image does *not* belong to a specific Manhattan like structure than sometimes the opposite (at two different stages in the segmentation). We The final segmentation is achieved using an MRF multi-label technique which provides a robust and precise framework for 2D and 3D fusion.

2.1 Estimating Manhattan Geometry

First we estimate the Manhattan System form the 2.5d data with a method that has been proposed by us in [9]. It is based on the idea to RANSAC a valid Manhattan system on the base of normal vectors. The method estimates the relative roll, pitch and yaw to the scene. The method also provides a plane segmentation of Manhattan-like structure, based on a 1D âĂŸâĂŹConnected Components” RANSAC that pre-segments all planes that are orientated to the Manhattan systems i.e. orientated to the X, Y or Z axis. For the sake of simplification, we assume only one system per scene.

2.2 Extracting 2D cues

The basic idea of our approach is to estimate per pixel the probability of the alignment (orientation of the pixel) to the one of the three vanishing points. Instead of using line segments we use straightforward gradients (i.e. orientation and magnitude). In the fashion of the Canny edge detector, we first blur the image using a Gaussian 3x3 kernel and apply a 3x3 Sobel filter on the image to obtain a gradient image. In a next step we calculate the reference orientation of each pixel to its three vanishing points as shown in figure 1 for two pixels. One can see that reference orientation of two vanishing points can be quite similar to each other, e.g. the sample point on the right bottom. In order to avoid artefacts in the estimation step, we calculate the similarity reference angles in a 3x3 matrix and use it as a additional gain in the estimation step. The estimation per gradient pixel to a vanishing point is a simple winner-takes-it-all method based (fig. 1(b)) on the smallest angle to a reference angle (of the vanishing point) i.e. we assume that every pixel is aligned only to one vanishing point. Finally we convert the value of the smallest angle into a probability by using a Gaussian weighting in a way that 5 degree difference will result in 3 σ.



(a) 2D Image with projected vanishing points (b) Colour coded alignments of gradients/pixels to vanishing points

Figure 1: Basic concept of vanishing points in 2D images. The vanishing points are colour coded for better visibility i.e. red=x, green=y and blue=z

2.3 Pre-Segmentation

Next, we use the plane hypothesis to pre-segment the individual voxels i.e. assign the voxels to planes and their orientation. This segmentation is done straightforward by re-projecting all hypotheses back into the 3D state space of the voxels. A voxel is assigned to a plane hypothesis if it intersects the plane hypothesis within a certain threshold. Each voxel is assigned only to one (or none, i.e. ”undecided”) ”best fitting” plane using the distance of the plane/voxel intersection to the mean of the voxels as the metric for matching. We count the number of inliers per plane similar to RANSAC. Planes with almost no support *count* $< 0.05\%$ are then removed from the set and corresponding voxels are freed.

In order to achieve a dense segmentation of the entire image we over segment the colour image using superpixels. The use of superpixels for over segmentation is quite popular in the computer vision literature within the last decade. The main objective is to locally merge pixels into ”superpixels” i.e. pixels with similar e.g. colour, texture, appearance or shading. In general we assume that true object boundaries are mostly (but not necessarily) represented by boundaries of the superpixels if the objects size is large enough (e.g. > 5 pixel). In this paper, we use the fast Minimum Spanning Tree based method by Felzenszwalb [10], giving us (by appropriate setting of parameters) 300-500 regions on average. However, any other over-segmentation method can be used.

2.4 MRF based multi-labeling

To label the pixels on a global level, i.e. to take into account prior information about possible plane orientations, 2D geometry and relations between neighbouring superpixels simultaneously, we formulate the problem in a fully probabilistic framework; as searching for a maximum posterior (MAP) configuration of the Markov Random Field [11] for multi-labeling [12]: In a labeling problem we are given a set of observations $\mathcal{P}$ (e.g. voxel and superpixel) and a set of labels $\mathcal{L}$ (e.g plane/orientation hypotheses). The goal is to assign each observation $p \in \mathcal{P}$ a label $f_p \in \mathcal{L}$ such that the joint labeling f minimizes the objective label function $E(f)$. We assume a graph $\mathcal{G} = \langle \mathcal{P}, \mathcal{E} \rangle$ consisting of a discrete set $\mathcal{P}$ of objects and a set $\mathcal{E} \subseteq \binom{|\mathcal{P}|}{2}$ of pairs of those objects.

An instance of the Max-sum problem is denoted by the tuple $(\mathcal{G}, \mathcal{L})$, where the elements $D_p(f_p)$, $V_{pq}(f_p, f_q)$ and $h_L \cdot \delta_L(f)$ of g are of alignment costs or qualities. The quality of a labeling f is defined

$$E(f) = \overbrace{\sum_{p \in \mathcal{P}} D_p(f_p)}^{\text{data cost}} + \overbrace{\sum_{pq \in \mathcal{N}} V_{pq}(f_p, f_q)}^{\text{smooth cost}} + \overbrace{\sum_{L \subseteq \mathcal{L}} h_L \cdot \delta_L(f)}^{\text{label cost}}$$

where h_l is the non-negative label cost of label l, and $\delta_L(f)$ is the corresponding indicator function

$$\delta_L(f) = \begin{cases} 1, & \text{if } \exists p : f_p = l \\ 0, & \text{else} \end{cases}$$

A common approach in the computer vision literature is to use three labels [5] for every major axis instead of using multiple labels per major axis. One reason to do so is that the general labeling for more than 3 labels leads often to NP-hard solutions [11, 12]. In this paper we use an MRF multi-label approach proposed by Delong et al. [12] which solves the multi-label problem within polynomial runtime for arbitrary number of labels. The polynomial runtime is achieved by using a different strategy for the multi-labeling than common approaches e.g. using a fixed number of labels: The main idea is to use the MRF to reduce the number of labels by merging them using the $E(f)$ function. The strategy includes starting with a reasonable number of labels e.g. all plane hypotheses and use the *smoothness term* as metric.

2.4.1 Graph entities

We build the graph $\mathcal{G}$ on the over-segmented image i.e. on the superpixels. The use of superpixels significantly reduces the number of objects in the graph compared to building the graph directly on the pixel grid. The superpixels represent objects (the set $\mathcal{P}$ in the graph and edges). The set $\mathcal{E}$, is established between every two neighbouring superpixels. The number of nodes (labels) K is set to the number of observed planes and four ”undecided” labels to mark ambiguous label assignments. The ”undecided” allow the solver to mark the places where there is *not enough*, information to decide which plane the superpixel belongs to. The individual ”undecided” labels are ”not_{xy}”, ”not_{xz}”, ”not_{yz}” and ”undecided”. The main idea with ”not” labels is that it is sometime easier to estimate where a plane orientation does not belong to than the opposite. The problem is that we have only sparse information at the scale of superpixel labels to determine the actual orientation of a label with a high certainty, like the typical scale space problem in computer vision. One can see that the estimation of the opposite is easier if we use a ”higher scale”. The ”not” labels are estimated using the 2D cues in the ”low scale” version and with the ”label cost” as the ”high scale” i.e. estimating the orientation of a set of labels.

We use an additional parameter to represent weights of edges that connect the superpixels. The calculation of superpixel likelihood is done using simple colour (RGB) histograms and using the Bhattacharyya [13] metric:

$$\rho[p, q] = \sum_{u=1}^{m} \sqrt{p_u q_u}$$

with $\sum_{u=1}^{m} p_u = 1$ and $\sum_{u=1}^{m} q_u = 1$ for u histogram bins of the superpixels p and q. Note that this implementation differs from many other MRF based label methods where the smoothness term is used to describe the likelihood of the neighbouring superpixels instead of using weights in the graph. With Delong et al. [12] approach the smoothness term describes the similarity between individual labels.

2.4.2 Smoothness term

The term $V_{pq}(f_p, f_q)$ describes the smoothness between two labels p, q i.e. the cost to assign q to p. The function itself must be injective i.e. $V_{pq}(f_p, f_q) \neq V_{qp}(f_q, f_p)$ which is necessary for our MRF variant by Delong et al. [12]. In our implementation we set cost function V_{pq} that a label q from a plane hypothesis is set to p of an another hypothesis, if both planes are assigned to the same orientation and they have almost the same position $|pq|$ (e.g. if the plane is y aligned, its the height) and if p has less support by voxels of the plane hypothesis, i.e. inliers, than q:

$$V_{pq}(f_p, f_q) = \begin{cases} 0, & \text{if } p = q \\ 1, & \text{if } p_{orientation} \neq q_{orientation} \\ cost_{pq} \cdot \Delta_{pq} \cdot q_{inliers}, & \text{if } p_{orientation} = q_{orientation} \\ cost_{known}, & \text{if } p = \text{”undecided”} \\ cost_{unknown}, & \text{if } q = \text{”undecided”} \end{cases}$$

with $cost_{pq} \gg cost_{unknown} > cost_{known} \geqq 1$. We use the distance of the plane position of the corresponding labels p, q as weight metric i.e.

$$\Delta_{pq} = \begin{cases} 2.0 - \frac{|pq|}{c} & \text{if } |pq| \leq c \\ 0 & \text{otherwise} \end{cases}$$

for a threshold c. We use $cost_{known} \geqq 1$ since we want to allow the MRF to remove false positives from the graph, i.e. false labeled x-axis oriented planes surrounded by z-axis oriented planes. The condition $p_{orientation} = q_{orientation}$ can combine labels with the same properties while the graph based representation ensures that this is only applied if the source superpixels are ”close” to each other in the source image.

2.4.3 Data term

The data term $D_p(f_p)$ encodes the quality of assigning a label f from the set $\mathcal{L}$ to an object/superpixel p in the graph. The quality measures how the superpixel is oriented to a specific plane. We use the pre-segmented labels of the voxels/pixel, i.e. the source pixels within the superpixels p. The data term for the "not" labels is given by the 2D cues. For each f_p we sum up the assigned (to vanishing points) gradient values (see figure 1(b)) to the corresponding labels i.e. "not_{xy}", "not_{xz}" and "not_{yz}". The idea is simple, if a gradient pixel is assigned to the z axis, it votes with its gradient value for the "not_{xy}" label. If the gradient is assigned to the x axis it votes for the "not_{yz}" label and y axis votes for the "not_{xz}" label. The overall assumption is that if a segment is assigned to a certain orientation, then it does to contain gradient pixels that are assigned to that axis, see figure 1. We normalize "not_{xy}", "not_{xz}" and "not_{yz}" with the total sum of the *raw* unassigned gradients within f_p. This avoids that false or few matches outvote the "assignment labels" e.g. within textured areas.

The data term for the "assignment labels" is given as follows: For each f_p we count the number of voxels/pixels with the same label as f from the pre-segmentation. Note that the labels f corresponds to the plane hypothesis. Next the number of pixels p_n is normalized to p_m and set as cost to $D_p(f_p)$

$$D_p(f_p) = \begin{cases} W(p_m) \cdot cost_{data}, & \text{if } W(p_m) \geq \lambda \wedge W(p_n) \geq \gamma \\ 0, & \text{otherwise} \end{cases}$$

With $cost_{data} \gg cost_{pq}$ and

$$W(p) = \begin{cases} p \cdot cost_{data}, & \text{if } f_p \text{ is "undecided"} \\ p, & \text{otherwise} \end{cases}$$

where λ and γ are thresholds and $cost_{data}$ is a normalizing constant that can prevent false positives if $cost_{data} > 1$. In our implementation we use $\lambda = 0.1$ and $\gamma = 10$ since we set the "minimum superpixel size" Felzenszwalbs [10] superpixels segmentation to 100 pixels. In our experiments $cost_{data} = 1.5$ which produces fewer, but more certain labels.

2.4.4 Label cost

The label cost term $h_L\dot{}\delta_L(f)$ is used to penalize each unique label that appears in f within $E(f)$. We use the cost h_L to express the certainty of the label f i.e. a lower cost reflects a higher certainty. Since we use the MRF to minimise the number of labels by fusing labels (=adding the costs of the fused labels), a solution will be used with minimal overall cost. We use the 2D cues to obtain h_L in a similar way we used it for the "not" labels, but using ratios instead of sums. For instance if a label is assigned to the x axis h_L is given as

$$h_L = \frac{sum_x}{sum_y + sum_z + \mu}$$

where *sum* is a function that sums all gradient pixels that are assigned to the corresponding axis and μ the smallest not-zero value that is assumable. The ratio for labels with y or z orientation calculates analog. Please note that we do not use the $\delta_L(f)$ as it was meant to be used in the first place as we distinguish only between three orientations of a plane and "unkown" labels with our 2D cues instead of the individual labels.

3 EXPERIMENTAL RESULTS

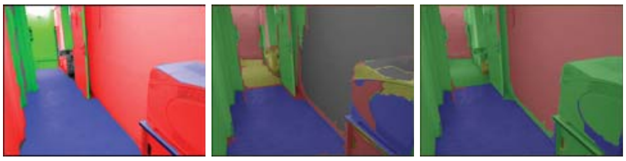
We choose a typical home environment (see fig. 2) for data acquisition using the Kinect. The data of all sensors is recorded at 25 frames per second. We recorded a representative set of six tours through our lab with a total length of approximately 320 meters. Three tours have a Manhattan like environment while the other ones represent a quasi Manhattan like environment.

Figure 2 shows sample images for all three tours and their segmentation in comparison with state of the art techniques from Saxena et al. [1] and Saxena et al. [1]. One can see that the combination of MRF and superpixels produces quite precise segmentation if stereo data is available and the superpixels do not contain glossy spots or other overexposed areas. In some case false-matches can appear if a glossy spot on the ground and the wall are to close too each other. The images also show that our parameterization of the segmentation is quite conservative since we want to produce only few false positives.

Our code runs on 2.4 GHz QuadCore PC, while the code is not optimized and uses only one CPU (except for MSAC Manhattan Geometry estimation). The average runtime for one frame is 408ms the clear bottleneck is the superpixel segmentation with 280ms. The next expensive function is the calculation of the Manhattan Geometry (80ms) due to the usage of histograms. Using smaller histograms will result in a lower constant runtime, but will also influence the accuracy negatively. The extraction of 2D cues the thrid third bottleneck with a constant runtime of 20ms followed by the mean shift clustering with 10ms.

4 CONCLUSION

In this paper we presented a novel robust method for room structure segmentation in a Manhattan like environment for 2.5D data using 3D and 2D cues. Once the camera



(a) our approach (b) Saxena et al. [1] (c) Hoiem et al. [2]

Figure 2: Segmented sample pictures from our test data: The colour indicates the alignment of a structure to an specific axis. The segmentation is executed on the identical superpixels on all three methods. Please note that the different approaches are using different color coding for the axis, yellow indicates uncertian planes in [1, 2].

orientation is estimated using MSAC, we calculate the assignment of every voxel to the 3 major axes and we extract plane candidates using histogram voting that are used as priors for a MRF based segmentation. We extract 2D cues from the 2.5D data using the concept of vanishing points and 2D image geometry. We also showed that the segemntion of the MRF can be improved using label cost metric and 2D cues.

The main drawback of our approach is that we depend on Manhattan-like structures which is common with many indoor environments. The extraction of 2D cues depends on the proper estimation of the camera orientation, however this could be overcome by estimating the vanishing points separately like in pure 2D computer vision approaches. Another drawback is that we depend on the output of the superpixel segmentation. Experiments have shown that the use of smaller superpixels is more robust than the usage of large ones, i.e. otherwise the superpixel segmentation tends to group the ground and white walls to one segment. At this stage we do not use multiscale oversegmentation, which would improve the performance of the 2D cues. Right now we kind of abuse the MRF as quasi multiscale by using the ”not” labels in a small scale and using it with the label costs in a large scale.

Our next step is to extend the approach with multi-scale over segmentation and to incorporate the 2D cues in the smoothness term of the MRF.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2011-2014) under grant agreement no FP7-ICT-2011-7 288146 (HOBBIT)

REFERENCES

[1] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3d: Learning 3-d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[2] D. Hoiem, A.A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 75(1), 2007.

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

[4] R. B. Rusu, A. Sundaresan, B. Morisset, K. Hauser, M. Agrawal, J. Latombe, and M. Beetz. Leaving flatland: Efficient real-time three-dimensional perception and motion planning. *Journal of Field Robotics, Special Issue: Three-Dimensional Mapping*, 26(10):841–862, 2009.

[5] B. Micusik and J. Kosecka. Piecewise planar city modeling from street view panoramic sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[6] D. Gallup, J. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[7] S. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *International Conference on Computer Vision (ICCV)*, 2009.

[8] A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *European Conference on Computer Vision (ECCV)*, 2010.

[9] Sven Olufs and Markus Vincze. Real time manhattanlik structure segmentation from kinect with constrained 1d cc-ransac. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR2011)*, 2011.

[10] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 2004.

[11] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(6):1068 –1080, jun. 2008.

[12] A. Delong, A. Osokin, H. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, California, 2010.

[13] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:564–575, May 2003.