CRF Based Frontier Detection using Monocular Camera

Sarthak Upadhyay
Robotics Research Center
IIIT Hyderabad, India

Suryansh Kumar
Robotics Research Center
IIIT Hyderabad, India

K Madhava Krishna
Robotics Research Center
IIIT Hyderabad, India

ABSTRACT

Frontier detection is a critical component in indoor mobile robot exploration, wherein the robot decides the next best location to move in order to continue with its mapping process. All frontier detection algorithms to the best of our knowledge require 3D locations of occupied regions as its input. In a monocular setting this entails a backend VSLAM algorithm that reconstructs the scene as the robot moves. Most monocular SLAM algorithms however provide sparse scene reconstruction from which frontiers cannot be reliably detected and estimated. In this effort we provide an alternate method of detecting frontiers during the course of robot motion that circumvents the requirement of dense mapping. Based on the observation that frontiers typically occur around vertical edges of walls, doors or tables we propose a novel linear chain CRF formulation that is able to detect the presence or absence of such frontier regions around such vertical edges. We used cues like increase in number of ground plane pixels and change in the spreading of optical flow vector, around those vertical edges. We also demonstrate that this method gives us more relevant frontiers as compared to methods based on reconstructing the scene through state-of-the art such SLAM algorithms such as PTAM. Finally, we present results in indoor scenes wherein frontiers are reliably detected around wall edges leading to new corridors, door edges leading to new rooms or corridors and table edges that opens up to a new space in rooms.

Keywords

Optical flow, Conditional Random Field.

1. INTRODUCTION

Frontier exploration was first introduced [11] as a means by which the robot expands its mapping process. While there have been several extensions and variants of it [2], [9], [4], all of them involved densely reconstructed maps of the scene with range sensors. Where vision has been involved, it has been an extension of frontier detection to a stereo setting [8] or through a combination of laser and monocular camera such as in [13]. The bottomline being the requirement of dense planar or spatial reconstruction of the scene has been inevitable in such approaches.

Herein, we present a novel method of frontier detection that operates over images bypassing the need for dense reconstruction with a monocular camera. Thus, this method can ideally complement a VSLAM backend [3] for expanding the mapping process without resorting to dense techniques such as [6]. We observe that frontiers typically occur around vertical edges in a scene such as doors, walls and tables. The proposed method tracks patches around such vertical edges and ascertains the spread of floor area and the spread of flow vectors over an image sequence obtained by the moving robot. A linear chain CRF uses the above spread to form its node and edge potentials and infers the presence or absence of frontiers around such edges. For example, vertical edges that do not lead to new areas around them such as closed doors are correctly detected as non-frontier regions (figure 1 depicts such situations). Extensive quantitative results that vindicate the efficacy of the proposed methods are shown in section 3. Specifically, the ability of the formulation to detect frontiers is shown to be largely invariant to the angles of approach of the robot to the edges. This is crucial for effective map building, for in realistic workspaces the robot's heading need not always be orthogonal to the normal of the plane that contains the edge. This is true even in case of corridors wherein due to odometry drift, the robot's heading is not always parallel to the corridor walls.

The approach closest to the current that uses monocular images for exploration appeared in [7]. Here, the frontiers were detected as the boundary of the segmented floor regions. The difficulty with such a formalism is that such frontiers could prove difficult to track over images and the ones that contain maximum information such as those leading to new corridors, rooms or openings can be potentially missed. Formulating the detection mechanism as a probabilistic state variable through probabilistic graphical models is another novelty of the proposed work.

2. SYSTEM OVERVIEW

Our method mainly consists of three basic modules. Firstly, LSD based line detection [10] to extract the vertical lines from an Image. Secondly, we use optical flow based tracking [1] to track feature points around the the extracted vertical lines and lastly, estimate the probability that whether the tracked vertical line is a possible frontier or not using Linear chain CRF (Section 2.5).

2.1 Pipeline

1. In the first step of algorithm, an image frame say I_n was subjected to Line Segement Detector algorithm [10], which actually detects a significantly huge number lines. However, we take only vertical lines that are touching the ground plane as non vertical lines do not serve any purpose of harboring frontier regions in an indoor environment.

2. Further, we track a patch containing features points, around those retrived vertical lines in the consecutive image frames(say for T frames). This helps us in determining the change in magnitude of optical flow vector around every region of interest, because a possible opening(a region to explore) will undergo a significant change in magnitude of flow vector over time.

3. In addition to this, we used ground plane segmentation algorithm [5], which adapts to the environment, and gives a good ground plane segmentation. So, we can easily keep track of change in number of ground plane pixels around each vertical lines obtained after step 1.

4. Lastly, we give these data obtained after step 2 and step 3 to Linear CRF framework to come up with a robust estimate of a frontier. A Linear Chain Conditional Random Field is formulated with its clique potential pertaining different states and observations.

2.2 Line Detection

Extremely homogenous environment always suffers from a featureless situations. With the advent of algorithms like LSD [10] we are able to detect lines even under extremely homogenous environmental conditions. LSD [10] algorithm gives us numerous line segments of an image. Of all these lines we consider only vertical or near vertical line segments as a possible candidates for frontier.

2.3 Feature Tracking

For tracking features around a vertical lines, we are using Large Displacement Optical Flow (LDOF) algorithm . Since indoor environment mainly consists of homogenous regions, it is indeed difficult to track sparse features around it. So, we are using a dense optical flow based tracking. However we find with LDOF [1] we are able to secure tracks reliable and effective enough to facilitate accurate frontier detection. While it is computationally expensive to densely track the entire image by tracking only few patches around vertical lines makes it a computationally feasible task.

2.4 Ground Plane Segmentation

Segmenting ground plane in an indoor environment suffers from various limitations like different texture of floor or it may have same color as of walls. Floor Segmentation algorithm described in [5], overcomes such difficulties and provides extremely satisfactory results for ground plane segmentation in an indoor environment. By using this algorithm it's easy to keep record of change in ground plane pixels over time.

2.5 Linear-chain Conditional Random Field

Linear chain Conditional Random Field (CRF) is an undirected graphical model, which considers the conditional dependencies among the random variables. This undirected graphical model is not confined to normalised conditional dependencies rather, its dependencies between the nodes can be any non-negative function. We pose our problem for frontier detection, as a distribution over hidden states of CRF by means of nodes and clique structures shown in Figure 1. Our model consists of two clique structure (s_t, s_{t-1}, s_{t-1}) and (s_t, z_t, z_{t-1}) respectively, for each time step t. In these structures, the conditional distribution over the hidden states $s_{0:T}$ factorizes into the product of clique potentials defined as:

$$p(s_{0:T}|z_{1:T}) = \frac{1}{Z(z_{1:T})} \prod_{t=1}^{T} \Psi_p(s_t, s_{t-1}, z_{t-1}).\Psi_m(s_t, z_t, z_{t-1}) \tag{1}$$

$$\Psi_p(s_t, s_{t-1}, z_{t-1}) = e^{\sum_{k=1}^{2} \lambda_k \cdot g_k(s_t, s_{t-1}, z_{t-1})} \tag{2}$$

$$\Psi_m(s_t, z_t, z_{t-1}) = e^{\sum_{k=1}^{2} \lambda_k \cdot f_k(s_t, z_t, z_{t-1})} \tag{3}$$

$g(.)$ and $f(.)$ is given by,

$$g(s_t, s_{t-1}, z_{t-1}) = p(s_t|s_{t-1}).p(s_{t-1}|z_{t-1}) \tag{4}$$

$$f(s_t, z_t, z_{t-1}) = p(z_t|s_t).\|h(z_t) - h(z_{t-1})\|^2 \tag{5}$$

Here, a state s_t can either be a frontier or a non-frontier $s_t = \{frontier, non-frontier\}$, z_t is the observation data. In our case we consider two observations ($k = \{1, 2\}$) that are increase in the number of ground plane pixels around a vertical line and spread of optical flow tracks around that vertical line over a time sequence. $k = 1$ correponds to increase in number of ground plane pixels and $k = 2$ considers spread of optical flow vector. $\Psi_p(.)$ contains the information about the transition of one state to another, given the prior state probability (equation 2) where as, $\Psi_m(.)$ holds the clue about the likelihold of observation given the state (equation 3) and change in the observation we get from previous observation. λ is a tuning parameter which was fixed after inferencing. $h(z_t)$ is an operator that computes the change in number of floor area around a patch discussed in 2.7.

2.6 Inference

In a CRF, sequences of output variables lead to enormous combinatorial complexity. A naive solution to this problem consists in marginalizing over all the state variables $S_1.....S_T$, which is equivalent to enumerating all the possible state sequences of length T. The computation has an $O(N^{\mathrm{T}})$ complexity, which, in most cases, is clearly infeasible. Thus, a dynamic programming approach is applied, known as the Forward-Backward Algorithm originally described for Hidden Markov Models. This algorithm can also be used for linear-chain Conditional Random Fields in a slightly modified form.

$$\phi(s, s', t) = \Psi_p(s, s', z_{t-1}).\Psi_m(s, z_t, z_{t-1}) \tag{6}$$

Forward pass,

$$\alpha_1(s) = P([S_1 = s]).P(z_1|[S_1 = s]) \tag{7}$$

$$\alpha_t(s) = \sum_{s'=1}^{2} \alpha_{t-1}(s').\phi(s', s, t) \tag{8}$$

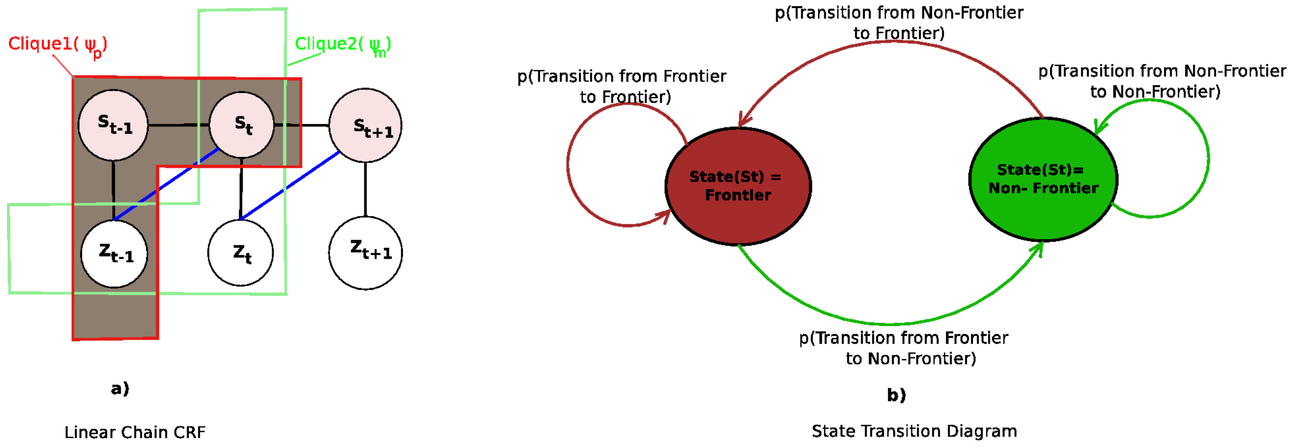

Figure 1: a)Shows the cliques of the graphical models used for our framework. b) Shows the state transition diagram that we used for frontier classification.

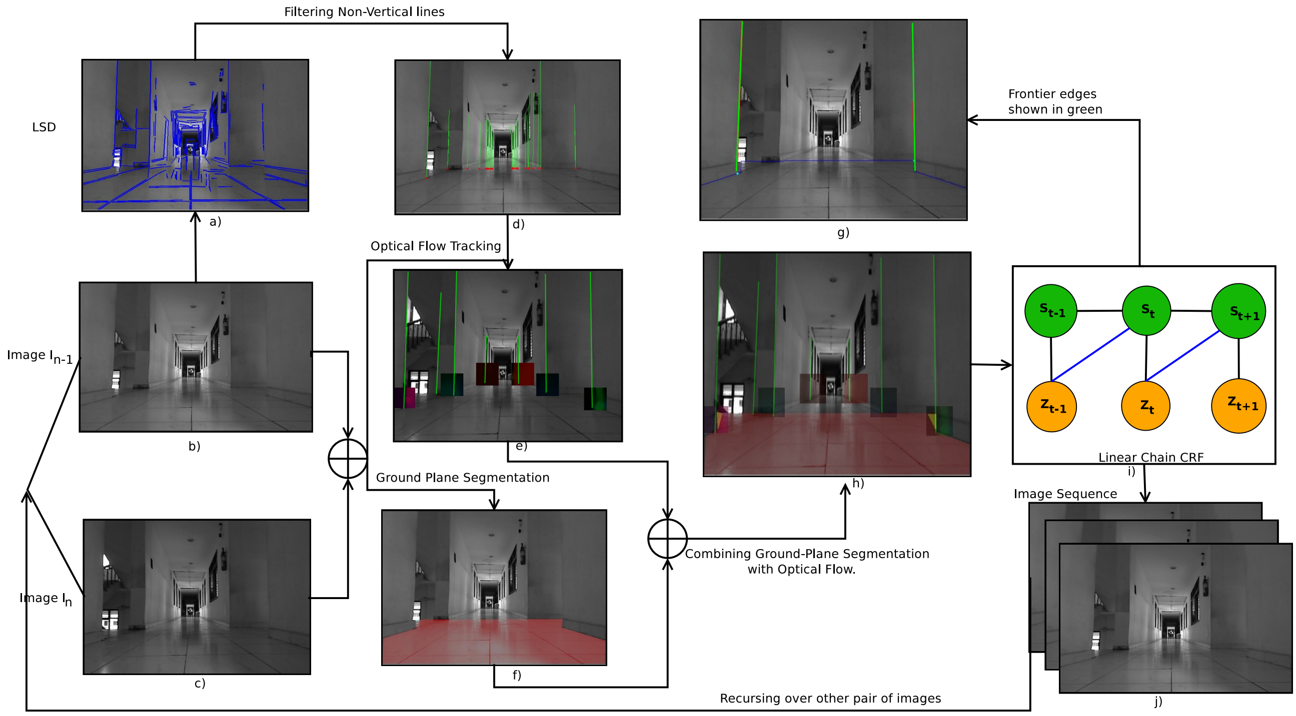

Figure 2: Flow Diagram of the proposed method. a) line detection using [10]. b), c) consecutive image pairs I_n, I_{n-1}. d) Filtered vertical or near vertical lines . e) dense optical tracks around the vertical line. f) Ground plane segmentation obtained by using [5]. g) final result of frontier detection. h) result obtained after combining e), f). i) CRF framework. i) Image sequence.

Figure 3: a)Result around wall edges leading to new rooms. b)& d)Result around table edges leading to new open area. c) Result around wall edges leading to new corridor.

Backward pass,

$$\beta_T(s) = 1 \tag{9}$$

$$\beta_t(s) = \sum_{s'=1}^{2} \beta_{t+1}(s').\phi(s, s', t+1) \tag{10}$$

$$Z = \sum_{s=1}^{2} \alpha_t(s) \tag{11}$$

State probability is computed as follows,

$$P([S_t = i]|z_{1\ldots t}) = (\alpha_t(i)\beta_t(i))/Z \tag{12}$$

The Forward-Backward Algorithm has a run-time of $O(S^2.n)$, so it is linear in the length of the sequence and quadratic in the number of states.

2.7 Our Approach

We come up with a very elegant approach to solve the problem of frontier detection using monocular camera. Given a set of consecutive images say $\{I_n, I_{n-1}\}$, we first extract possible vertical lines in the image I_{n-1} using [10]. After the vertical line are extracted faithfully, we find dense optical flow track in the subsequent image I_n around all vertical lines extracted from the image I_{n-1}. The same set of images were also used to obtain ground plane segmentation [5].

The key idea behind doing this preprocessing step for frontier detection is that, if there is a place for exploration we should get a significant change in the magnitude of optical flow vector around vertical lines. Figure 2e) shows the dense flow vector obtained after tracking the features in the consecutive images. The color changes indicates the change in optical flow vector. It can be inferred from this figure, that there is a steep color changes around some of the vertical lines. Relying on flow vector cues alone could pose certain other challenges. For example, flow vectors could be substantially different around window edges, which from a robotic setting do not harbor frontiers despite giving a perception of new areas through the window grills. Hence, we also resort to change in the spreading of ground regions around vertical edges as another cue for confirming the presence of frontier regions.

For estimating the spreading of ground pixels the following is done. A number of patches are selected around a vertical edge and the extent of the floor in those patches ascertained. Based on robot odometry and ground plane normal the floor regions in those selected patches are warped into the next view of the robot and the area of the floor expected to be seen is predicted. The difference between the observed floor area and the predicted warp is computed for each tracked patch and averaged over the number of patches. This averaged difference constitutes the second term in the right hand side of equation 5.

Combining the two data term, we get a substantially strong evidence for classifying a vertical line as a frontier or not. But it is quite obvious that, the change of ground plane pixels and optical flow vector around the vertical lines cannot be observed quickly, it requires an accrual of such observations over time. Since relation between observations are critical for accurate detection, we pose the problem as a linear chain CRF that is able to capture relations between observations as well as states. A liner chain CRF caters to both the measurement potential and state change potential information overtime Figure 1a). We also used a transition probability distribution obtained empirically i.e $p(s_t|s_{t-1})$ shown in Figure 1b).

Using the CRF framework to capture the relation between states as modeled in equation 4 and 5, we are able to obtain good frontier detection over time. Lets us consider an example to explain why our formulation works in such conditions. Intially, prior probabilities were kept equal $p(s_o|z_o) = 0.5$ for both states $s_t = \{frontier, non-frontier\}$. We obtained likehood of observation i.e $p(z_t|s_t)$ and state transition matrix $p(s_t|s_{t-1})$ by training data set. Now suppose we are tracking a vertical line over a T time horizon. $\Psi_p(.)$ is an indicator of how likely you will observe a state s_t given prior state probability and transition probability. So, at $t = 0$ we will not get much information about how likely it a frontier or not. Whereas, $\Psi_m(.)$ gives us a information about what is the likehood of observation given the state. So, if the change in observation i.e $\|h(z_t) - h(z_{t-1})\|^2$ is approximately same for both states, it automatically gets corrected by $p(z_t|s_t)$. Here, $h(z_t)$ computes the observed floor area of the current view, whereas the $h(z_{t-1})$ computes the predicted area of the warp based on the previous patch, current robot pose obtained from odometry and the normal to the ground plane. The potential of these two cliques is calculated recursively for each vertical lines in the subsequent image frames for T time horizon, thereby updating $p(s_t|z_t)$ after each iteration. Hence, resulting in a better state estimate.

3. RESULTS

For our experiments, we use Kinect mounted on a iCreate TurtleBot. Kinect is used only as a monocular camera and not as depth sensor. The whole algorithm is processed on a laptop connected to robot. Laptop runs on a 2.3 GHz Quadcore i7 processor.

We provide insights into the working of the algorithm through the following example (figure 3a - 3d). Figure 3c shows the Turtlebot moving along a corridor approaching another corridor that runs orthogonal to it. The CRF formulation is expected to detect the two vertical edges that leads to new areas into the left and right as harbouring frontier regions. The vertical edges that remain after filtering are shown in green. Initially it is difficult to estimate the state of these edges or equivalently the state of the areas around these edges. As the robot navigates the difference in flow vectors around the left and right edges become prominent as shown in Figure 2e. Concurrently, the ground plane areas spread around the left and right edges prominently making the corresponding potentials active (figure 2h), resulting in correct detection of frontier regions. Figure 2g shows the edges that harbor frontiers in green while non frontier edges are filtered after CRF has run its course over T frames.

Results for doorways and desks are shown in figures (3d - 3d) and (4b). Figure 4b shows the door edge amongst other edges at the first view. Figure 4c shows the result of the CRF based state estimation. The door edge is correctly estimated as the frontier edge while all others are estimated to be non frontier edges. Figure 4a shows the edge of a closed door amongst other edges. That the edge was correctly estimated to be a non frontier edge is shown in Figure4c thereby vindicating the efficacy of the method to distinguish closed and open doors. Figure 3b and 3d show identical results for a table edge around which new areas to be explored reside.

Table 1-2 depicts quantitatively the performance of the proposed formulation. Three most typical and dominant frontier edges that occur in indoor scenes were considered. These constitute the rows of the table. The scenes were also composed of a number of other non frontier edges. Each major column of the table indicates the various angles at which the robot approached the edges. These are at orientations of 15, 30, ... Each such major column is composed of two sub columns, one depicting the false positive percentage and the other false negative percentage. These were computed by averaging over several such runs of a robot made to move realistically in an indoor setting. The low percentage of false positives and negatives confirm the efficacy of the formulation.

4. COMPARISON WITH OTHER METHODS

Parallel Tracking And Mapping(PTAM) and other Structure from Motion(SfM) techniques forms a very sparse map of an indoor environment due to lack of trackable features. This map often leads to wrong frontier selection, subsequently leading to an awful exploration, some of them leads the robot into walls. Figure 5c) shows that PTAM map gives frontiers leading into walls. Due to lack of texture, the walls are not mapped and are shown as openings for further exploration. This shows the inability of standard frontier methods to detect reliable frontiers from a sparse map reconstructed by the popular SLAM framework. However, our approach gives a faithful frontiers in all such situations. Thereby, preventing the robot from several risks such as a) exploring the same region again and again, b) crashing into walls due to spurious frontier selection etc.

5. CONCLUSIONS

A novel linear chain CRF formulation for estimating frontiers for a mobile robot equipped with a monocular camera was presented and verified to work reliably and robustly. In indoor scenes consisting of frontier edges from tables, doorways and walls the performance was found to be confirming its accuracy and soundness. Frontier detection is critical to automate a SLAM backend as the robot decides the next best location to move in order to expand its mapping process. Almost all the frontier detection algorithms known to the authors work with densely reconstructed scenes (planar or spatial) obtained from laser range finders or stereoscopic systems. To the best of the author's knowledge this is the first such frontier detection detailed to work with monocular camera. The authors believe that this would serve to advance the state of the art for automating monocular SLAM systems, wherein the reconstructed scene is typically too sparse for accurate detection of openings and frontiers. The paper also shows the inability of standard frontier methods to detect frontiers from a sparse map reconstructed by the popular SLAM algorithm, PTAM. From that point of view as well the proposed method can be considered useful.

6. REFERENCES

[1] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):500–513, 2011.

[2] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider. Coordinated multi-robot exploration. *Robotics, IEEE Transactions on*, 21(3):376–386, 2005.

[3] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[4] A. K. Krishnan and K. M. Krishna. A visual exploration algorithm using semantic cues that constructs image based hybrid maps. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1316–1321. IEEE, 2010.

[5] S. Kumar, A. Dewan, and K. M. Krishna. A bayes filter based adaptive floor segmentation with homography and appearance cues. In *Proceedings of*

a) b) c)

Figure 4: a) spread of ground plane pixels when door is closed. b) optical flow spread around the frontier and spread of the ground plane pixel at the door opening. c) frontiers(green) and non-frontier(brown) obtained after using CRF.

a) Image b) Sparse 3D map c) Detected Frontiers

Figure 5: a) The original environment. b)3d reconstruction using PTAM [3]. c) Ground map of the environment with wrongly detected frontier shown by arrow mark.

Frontier Type	0°		10°		20°		30°	
	FP(%)	FN(%)	FP(%)	FN(%)	FP(%)	FN(%)	FP(%)	FN(%)
`Corridor Edges`	0.8	5.0	0.8	0.2	0.6	0.0	0.8	0.0
`Door Edges`	0.4	5.2	0.6	0.6	0.4	0.2	0.4	0.2
`Desk Edges`	0.8	4.8	0.6	0.8	0.6	0.4	0.6	0.2

Table 1: Analysis with respect to Angle of Approach at 2m

Frontier Type	0°		10°		20°		30°	
	FP(%)	FN(%)	FP(%)	FN(%)	FP(%)	FN(%)	FP(%)	FN(%)
`Corridor Edges`	1.2	6.3	0.9	0.4	0.4	0.2	0.6	0.0
`Door Edges`	1.5	6.0	0.3	0.3	0.2	0.4	0.4	0.1
`Desk Edges`	1.2	5.8	0.7	0.6	0.8	0.4	0.3	0.4

Table 2: Analysis with respect to Angle of Approach at 3m

the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, page 54. ACM, 2012.

[6] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327. IEEE, 2011.

[7] D. Santosh, S. Achar, and C. Jawahar. Autonomous image-based exploration for mobile robot navigation. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2717–2722. IEEE, 2008.

[8] R. Sim and J. J. Little. Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2082–2089. IEEE, 2006.

[9] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *AAAI/IAAI*, pages 852–858, 2000.

[10] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A line segment detector. *URL http://www. ipol. im/pub/algo/gjmr line segment detector*, 2012.

[11] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pages 146–151. IEEE, 1997.

[12] R. I. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge University Press, second edition, 2004.

[13] D. Maier, M. Bennewitz, and C. Stachniss. Self-supervised obstacle detection for humanoid navigation using monocular vision and sparse laser data. In *IEEE International Conference on Robotics and Automation (ICRA), May 2011, pp. 1263 âĂŞ1269*