Growing semantically meaningful models for visual SLAM.

Anonymous CVPR submission

Paper ID ****

Abstract

Though modern Visual Simultaneous Localisation and Mapping (vSLAM) systems are capable of localising robustly and efficiently even in the case of a monocular camera, the maps produced are typically sparse point–clouds that are difficult to interpret and of little use for higher–level reasoning such as obstacle avoidance or navigation. In this paper we begin to address this deficiency, presenting progress on expanding the competency of visual SLAM systems to build richer maps. Specifically, we concentrate on modelling indoor scenes using semantically meaningful surafces and accompanying labels, such as "floor", "wall", and "ceiling" — an important step towards a representation that can support higher-level reasoning and planning.

We leverage the Manhattan world assumption and show how to extract vanishing directions jointly across a video stream. We then propose a guided line detector that utilises known vanishing points to extract extremely subtle axis–alligned edges. We utilise recent advances in single view structure recovery to building geometric scene models and demonstrate our system operating in real–time.

1. Introduction

The simultaneous localisation and mapping problem has recieved considerable attention over past decades, which is unsuprising given its centrality in fields from mobile robotics to augmented reality. Considerable progress has been made over this period and modern SLAM systems perform efficiently and robustly even in the case of a monocular video stream [8].

Many high–level reasoning and planning problems can benefit from an accurate underlying SLAM system, but SLAM point clouds alone provide a poor basis upon which to reason about scene semantics as they represent just a fraction of the information present in the original images. Photometric cues for edges, surfaces, occlusion boundaries, and texture information, among others, are lost entirely.

An important step towards higher–level reasoning tasks is to represent the structure of the scene at a semantic level, using meaningful concepts such as "floor" and "wall". These entities assist in reasoning as they correlate with, for example, the locations in which objects might appear or the actions that might be feasible in various locations.

The present work investigates the extraction of such a semantic scene model from video sequences, utilising an underlying SLAM system together with rich photometric cues. We focus on indoor scenes as they exhibit a rich set of regularities that assist in model building. One such regularity is the prevalence of three mutually orthogonal orientations around which man–made environments are often built up. The use of this observation to restrict the space of possible scene interpretation has come to be known in the literature as the Manhattan–world assumption. Typically, the orientation of the camera with respect to these dominant directions is *a priori* unknown and must be extracted explicitly, for example by identifying vanishing points. We propose a new method that leverages the camera poses provided by SLAM to estimate vanishing directions jointly across a video sequence, allowing frames with salient edge information to inform the system about vanishing directions in frames with poor or non–existent edge information.

We then return to the images and re–identify line segments using the known vanishing points to inform our search. We propose a novel line detector that takes vanishing point locations into account to identify important structural edges that are aligned with a dominant direction but which may exhibit weak gradient magnitudes, whilst ignoring stronger gradients generated by surface texture or occlusion boundaries.

The final component of our system joins the line segments into Manhattan building structures, inspired by recent work in single view reconstruction [3]. We enumerate possible building structures that the observed line segments could generate under the Manhattan world assumption, evaluating each for consistency with surface orientations estimated from photometric cues. The key contribution of this section is the extension to multiple views of the hypothesis testing framework, which we show gives significantly improved results.

The remainder of this paper is organised as follows. In

the next section we overview prior work in this field. Following this we describe the three primary components of our system in order: the joint vanishing point estimator, the guided line search algorithm, and the reconstruction system, with results given in each section. Finally we discuss the results and present closing remarks.

2. Background

In recent years the need for semantics to be connected with SLAM maps has been recognised by several researchers. Stachniss *et al*. [14] have taken an image–centric approach wherein 3D features are projected into frames and used together with photometric cues to classify environments into semantic categories such as "corridor" or "room". Posner *et al*. [10] take this a step further by segmenting incomming frames based jointly on 3D and photometric cues, after which image segments are assigned labels such as "grass" or "vehicle". Xiao and Quan [17] have aproached this problem by solving a multiple label MRF over superpixels from two or more views, with 2D and 3D cues combined to form node potentials. Brostow *et al*. [1] also combined photoemtric and geometric cues but used simpler cues and did not force consistency across views.

Buschka and Saffiotti [2] opt instead to reason directly in the map. They build an occupancy grid and identify room boundaries by repeated application of mophological filters. More recently, Golovinskiy *et al*. [5] learn to segment and identify objects in city–wide reconstructions using machine learning techniques. These approaches discard the original images after building a map, which we believe discards many useful cues not captured in the map representation.

Furukawa *et al*. [4] have shown how to produce reconstructions of indoor scenes under the Manhattan–world assumption. Their reconstructions are of very high quality but they quoted computation times of between one minute and one hour forty minutes, which makes their approach unsuitable to our on–line situation. Their system uses multiple–view stereo to first reconstruct a dense point cloud, whereas we are interested in working with a sparse point cloud and leveraging photometric cues for real–time performance.

Several authors have recently demonstrated impressive single view reconstruction systems. Hoiem *et al*. [7] pose the problem as a multi–class segmentation problem, with labels corresponding to 3D geometry. Saxena *et al*. [12] obtain reconstructions by estimating surface normals of image patchlets. Gould *et al*. [6] label with both geometry and object classes for improved accuracy.

Lee *et al*. [3] take a geometric approach in which detected line segments are connected to form hypotheses about 3D structure. The authors show that the Manhattan worlds highly constrain the set of possible building structures, so an exhaustive search is possible. This work forms the basis for Section 5 in which we extend this approach to multiple views.

Many researchers have proposed methods for completing partial lines or identifying subtle–yet–important line segments that humans see easily. For example, Sarti *et al*. [11] iteratively fill missing boundaries starting from a reference point, Tuytelaars *et al*. [16] iterate between detecting lines and identifying intersections using the Hough transform, and Shufelt [13] models line detection errors explicitly. This body of literature deals with the single image case, whereas we use known vanishing points obtained from multiple views to improve the accuracy and speed of the search.

3. Extracting a canonical coordinate frame

In order to make use of the Manhattan world assumption we must first discover the orientation of the Manhattan world with respect to the camera. Equivalent to the Manhattan world assumption is the statement that there exists a "canonical" coordinate frame in which world surfaces are axis–aligned. The problem is therefore to discover the rotation R_w mapping canonical coordinate to SLAM coordinates. Since our SLAM system has already determined the relative poses between successive frames, this rotation is fixed for all frames.

In the past researchers have discovered R_w for single images by identifying vanishing points (cite), or in the structure–from–motion setting by clustering surface normals estimated from local neighbourhoods of point clouds (cite Furukawa). The former approach fails for frames containing edges in only one or none of the three dominant directions — a common occurence for video sequences of indoor environments — and for frames containing many edges not aligned with one of the dominant directions. Since R_w is fixed for all frames it makes sense to leverage all available data during estimation rather than to estimate vanishing points seperately for each frame. The surface normal approach requires a dense scene reconstruction, which is currently prohibitive in for real–time applications.

We begin by sampling a set of key frames from the input sequence with the number of key frames chosen for tractibility. On each key frame we run the Canny edge detector followed by a standard edge linking algorithm to identify a set of straight line segments $L_i = \{\boldsymbol{x} : \boldsymbol{l_i}^T \boldsymbol{x} = 0\}$ and associated confidences c_i, computed from the ratio of eigenvalues of each line segment's constituent pixels.

The SLAM system provides a pose P_i for each frame, which contains a rotation R_i and translation $\boldsymbol{t_i}$. These are measured with respect to some arbitrary coordinate frame determined during initialization, which we will henceforth refer to as the "SLAM" coordinate frame.

In the canonical coordinate system the vanishing points have homogeneous coordinates $(1,0,0,0)$, $(0,1,0,0,0)$, and $(0,0,1,0)$. Their projections into camera i are given

by

$$\boldsymbol{v_x} = R_i R_w \boldsymbol{e_x} \quad (1)$$
$$\boldsymbol{v_y} = R_i R_w \boldsymbol{e_y} \quad (2)$$
$$\boldsymbol{v_z} = R_i R_w \boldsymbol{e_z} \quad (3)$$
$$(4)$$

where $\boldsymbol{e_x}$, $\boldsymbol{e_y}$, and $\boldsymbol{e_z}$ are unit vectors in the x, y, and z directions respectively, and R_w is the mapping from canonical coordinate to SLAM coordinates.

We can now write down an error function to be minimised in terms of the world rotation R_w:

$$E(R_w) = \sum_i \sum_j \sum_{k=1}^{3} r_{jk} (l_j^T \; R_i R_w e_k)^2 \quad (5)$$

where r_{jk} is the responsibility of the k^{th} vanishing point for the j^{th} line segment. The term $R_i R_w e_k$ represents the projection of the k^{th} vanishing point into the i^{th} frame, and $(l_j^T \; R_i R_w e_k)^2$ is the deviation of the j^{th} line segment from it. The full error function (5) is then a sum over all line segments and vanishing points, with the deviations weighted by the estimated responsibilities of each vanishing point for the line segments.

While other authors search for vanishing points by clustering on the Gaussian sphere, enforcing orthogonality constraints afterwards, we prefer to optimise in terms of R_w directly, which builds the orthogonality constraints into the core of the estimation process.

We now proceed to describe our EM algorithm. Our expectation step computes the responsibilities of the vanishing points for the line segments. We assume a Gaussian likelihood

$$p(l_j \mid v_k) = G(l_j^T v_k; \mu) \quad (6)$$

as well as a fixed prior on observing a spurious line segment

$$p(s) = \rho \quad (7)$$
$$p(l_j \mid s) = 1 \quad (8)$$

The responsibilities are then computed as the normalized likelihoods

$$r_{jk} = \frac{p(l_j \mid v_k)}{\rho + \sum_k p(l_j \mid v_k)} \quad (9)$$
$$(10)$$

The M step consists of optimising R_w with respect to the error function (5). There is no closed form solution for the optimal R_w so we instead perform gradient descent. We represent R_w in the Lie algrabea as a member of the special orthogonal group $SO(3)$.

$$R_w = exp(\sum m_i G_i) \quad (11)$$

where the G_i are the generator matrices for $SO(3)$ and m_i are the respective coefficients. The advantage of using the Lie algebra is that at each step we are guaranteed that R_w remains a pure rotation, whereas in other representations, such as simply optimizing the elements of the 3×3 rotation matrix, this is not the case.

We now differentiate (5) with respect to the coefficient vector $\boldsymbol{m}$:

$$\nabla f(x) = \sum_i \sum_j \sum_{k=1}^{3} (2 r_{jk} l_j^T R_i R_w e_k \;\; r_{jk} l_j^T R_i \nabla R_w e_k) \quad (12)$$
$$\nabla R_w = [G_1 \boldsymbol{e_1}, G_2 \boldsymbol{e_2}, G_3 \boldsymbol{e_3}] \quad (13)$$

In accordance with the standard gradient descent approach our update rule is then

$$\boldsymbol{m^{t+1}} = \boldsymbol{m^t} - \frac{f(\boldsymbol{m^t})}{\|\nabla f(\boldsymbol{0})\|_2} \nabla f(\boldsymbol{0}) \quad (14)$$

In summary, to obtain R_w we iterate between assigning responsibilities (the E step) and optimising the rotation between world and SLAM coordinates (the M step). Each M step consists of a gradient descent in the Lie algebra. In practice we found that our system converges in around 25 iterations of the EM algorithm, and around 20 steps are required for each gradient descent.

Figure 1 shows the vanishing points identified in one of our sequences. Since each frame is informed by the entire sequence we are able to identify a globally consistent coordinate frame where single–image vanishing point detection fails. Figure 2 shows a side–by–side comparison with the single–image vanishing point detector of [9]. Recently proposed improvements to single–image aproach [15] may improve slightly on these but we found that in cases where the single–image approach fails there is often simply not enough information available in individual frames to identify the appropriate coordinate frame, so any single–image approach will necessarily fail.

3.1. Identifying the Vertical Direction

Of the three axes in the canonical coordinate frame, the one corresponding to the vertical direction is semantically distinct from the others since it defines the orientation of the ground and ceiling planes, and also the direction in which gravity operates. It is easy to identify this axis since humans necessarily move over the ground plane when capturing a sequence, and have limited scope for moving the camera in the up–down direction. We therefore set the vertical direction to that in which the range of camera poses is smallest. Having identified R_w there are only three possible choices for the vertical direction and we found this heuristic to work effectively in all of our sequences.

An alternative approach would be to make use of the tendency for humans to orient the camera so that the vertical

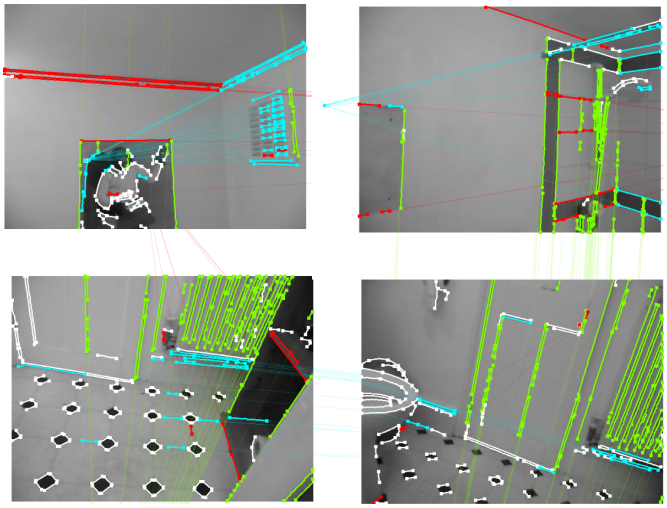

Figure 1. Four frames from the "bathroom" sequence and the detected vanishing points. The vanishing points are correctly identified despite the strong distractor gradients generated by the floor tiles, which is possible only by integrating information from multiple views into the estimation process. Also note that the up–down, east–west, and north–south directions are labelled consistently in each frame.

direction in the world corresponds to the up–down direction in the image. The major drawback of this approach arises in the common case that an entire sequence is filmed with the camera rotated by 90° so as to capture a high, narrow environment, in which case this test fails.

3.2. Relaxing the Manhattan world assumption

The strong Manhattan assumption states that any pair of surfaces of interest are either parallel or orthogonal to one another. Amongst indoor scenes, one common deviation from this is scenes with vertical walls that are not pair–wise orthogonal. We define the weak Manhattan assumption as "the environment consists of a horizontal ground plane and corresponding ceiling plane, and a set of vertical wall segments extending continuously between them." This formulation permits pairs of wall segments that are not orthogonal to one another.

Weakly Manhattan scenes contain much of the regularity of strongly Manhattan scenes. We deal with the weak Manhattan assumption as follows. First, we run the rotation recovery algorithm described above. Next, for each line marked as spurious by the EM algorithm we find its intersection with the horizon,

$$\boldsymbol{v}_+ = R_w^{-T} R_i^{-T} \boldsymbol{l_j} \times \boldsymbol{e_z} \tag{15}$$

which would be its vanishing point if it were horizontal. The horizontal lines on all vertical wall segments in with a given orientation will coincide in their intersection with the horizon. We parameterize each such intersection according

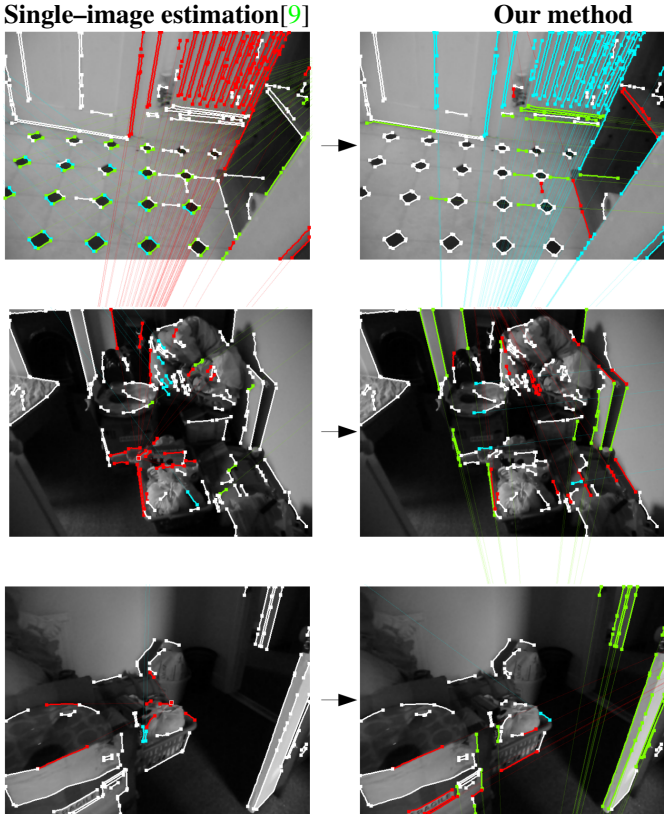

Figure 2. Comparison between vanishing points estimated for single views (left column) and joint estimates from 20 frames in a video sequence (right column). Our method is able to identify vanishing directions correctly in these difficult cases, whereas the single view estimator is confused by non–Manhattan line segments.

to the angle it makes with the x and y axes

$$\theta_j = \text{atan}(\boldsymbol{e_y}^T \boldsymbol{v}_+, \boldsymbol{e_x}^T \boldsymbol{v}_+) \tag{16}$$

We accumulate the θ_j into histogram bins and identify any bins above a fixed threshold k that are local maxima. Such bins represent the orientation of the additional vertical wall segments we seek. Finally, we re–estimate the vanishing point for each orientation by minimising the likelihood described in the previous section.

4. Directed Line Search

Many of the structurally important edges in indoor scenes generate gradients far weaker than those generated by surface texture or occlusion boundaries. For example, in the common scenario that the walls in a room are painted the same colour, the image gradients at wall intersetions will be generated only from subtle lambertian lighting differences.

We have found that the standard line detection approach (Canny followed by edge linking) is incapable of detecting many important structural edges unless the thresholds are

lowered to such a point that any textured surface generates many thousands of spurious line segments. However, if the Canny thresholds are set to some level high enough to reduce spurious detections to a manageable level then the true positive line segments are plentiful enough to robustly detect vanishing directions but not to recover building structures (as described in the following section).

To overcome this, we return to the images after determining the vanishing directions and search again for line segments, this time with the known vanishing directions informing the search.

Each vanishing point is associated with a one–parameter family of lines extending from it. Parameterizing this family is non–trivial since there exists a continuum of scenarios from parallel lines meeting at infinity to lines radiating outwards from a vanishing point inside the image bounds (see Figure 3). We choose to parameterize the lines by the angle θ they form in the plane generated by the other two vanishing points

$$\theta(i, \boldsymbol{x}) = \text{atan}(\boldsymbol{x}^T \boldsymbol{v_j}, \boldsymbol{x}^T \boldsymbol{v_k}) \tag{17}$$

where all vectors are in homogeneous coordinates. This representation is free of singularities and sampling uniformly in θ space produces roughly uniformly distributed lines in the image.

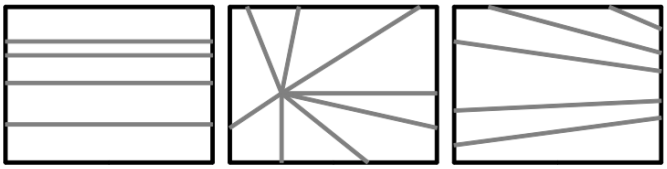

Figure 3. Lines meeting at a vanishing point.

For each pixel $\boldsymbol{x}$ we begin by estimating its vanishing point association. Since vanishing points are always well–seperated (due to orthogonality) we obtain a reasonable estimate of the vanishing point a line containing the pixel would be associated with from the image gradient $\boldsymbol{g_x}$.

$$\text{assoc}(\boldsymbol{x}) = \text{argmin}_i((\boldsymbol{x} - \boldsymbol{v}_i)^T \boldsymbol{g_x}) \tag{18}$$

where i ranges over the three possible vanishing points.

Next we build a histogram over θ for each of the three vanishing points. Pixels vote for the lines they generate according to the strength of the gradient at the pixel and the agreement between the pixel's gradient orientation and the direction to its associated vanishing point. Formally, the weight with which a pixel at $\boldsymbol{x}$ votes for the line $\boldsymbol{l}_{i,x}$ it generates with the i^{th} vanishing point is given by

$$w_{i,\boldsymbol{x}} = (\|\boldsymbol{g_x}\|_2)^\alpha \ \cos^\beta((\boldsymbol{x} + \boldsymbol{g_x})^T \boldsymbol{l}_{i,x}) \tag{19}$$

where α and β are parameters that determine the relative importance of the gradient orientation and magnitude.

The bin width for the histograms is set to the minimum size such that no bin spans more than two pixels anywhere in the image. Histogram peaks are identified by applying non–maximum suppression followed by thresholding. The final line segments are identified by walking along lines corresponding to the identified histogram peaks and linking edge pixels using hysteresis. A line segment is started each time a gradient magnitude above the high threhold k_{high} is detected, and is ended when the gradient magnitude drops below the low threshold k_{low}.

This algorithm is highly efficient since only one pass over the image is needed to build all three histograms. Identifying histogram peaks and walking along the corresponding lines is then computationally trivial.

Figure 4 shows four example frames and the lines corresponding to the histogram peaks. The detector fires at subtle axis–aligned gradients while ignoring strong but non–axis–aligned segments. Figure 5 shows a side–by–side comparison with the line detector of [9], which employs the standard Canny edge detector followed by an edge linking algorithm. In each example our detector is able to identify important structural edges that the Canny detector does not respond to. We found that lowering the Canny thresholds sufficiently that these edges were detected generated many thousands of spurious line segments on the textured carpet and other areas.

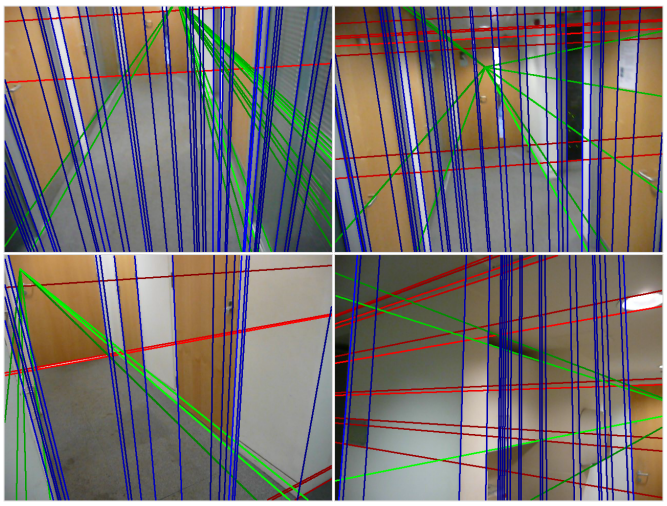

Figure 4. Four frames from the "lab" sequence, with peaks in the the histograms over θ highlighted. The rays capture the important geometric structure extremely accurately, with almost no false positives.

5. Recoving Building Structures

The final component of our system estimates building structures from the set of axis–aligned lines produced by our line detector. Lee *et al.* [3] have shown that under as-

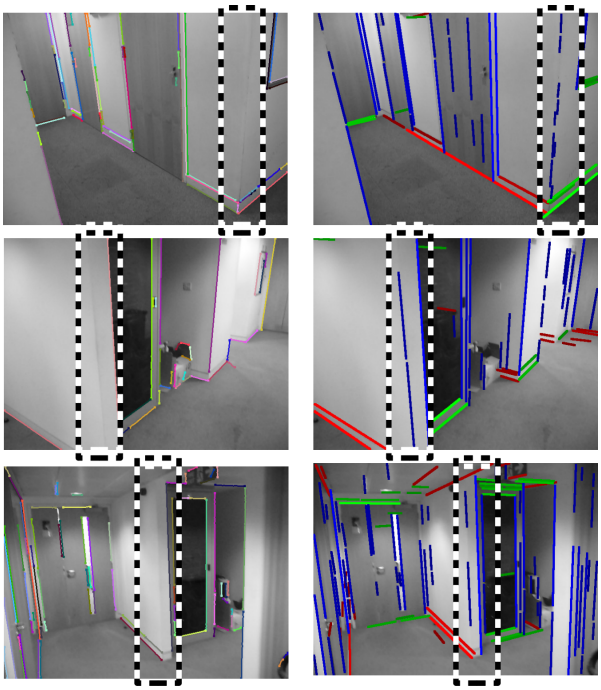

Figure 5. Comparison between our guided line search (right–hand column) and the Canny/edge linking detector (left–hand column). Our system is able to recover several subtle yet structurally significant line segments that Canny misses.

sumption of a Manhattan world containing infinite floor and ceiling planes between which vertical wall segments extend, there are relatively few possible intersections to consider. Floor/wall intersections and ceiling/wall intersections generate line segments with vanishing directions on the horizon. Vertical line segments arise either when two wall segments meet or when one wall segment occludes another. In the latter case the possible vanishing directions of the wall segments on the left and right of the occlusion are constrained by the position of the intersection in the image. With these constraints, the entire set of building structures can be enumerated by branch–and–bound. For further details, the reader is referred to [3].

Lee *et al*. evaluate hypotheses by checking for consistency with pixelwise orientation estimates in the image. We extend this to use multiple images for hypothesis evaluation. A building structure B defines a unique 3D model up to an unknown scale factor s^*. Lee *et al*. ignore this parameter since they build reconstructions for single–images only. However, in our case s^* is required to transfer the model between frames.

To determine s^*, we leverage the observation that some map points will fall on the surfaces we are trying to reconstruct. We therefore devise the following voting scheme. For each map point visible in the current frame we identify the surface it falls upon within the hypothesized building structure. Next we compute the scaling s^- such that the reconstructed surface contains that 3D point. Each map point then votes for the scale it induces on the building structure, where the votes are discretized to allow efficient vote counting. To avoid patches of detailed texture dominating other cues, each surface is allowed to vote at most once for any s^*.

Knowledge of s^* permits construction of a full 3D model and hence allows transfer of the building structure between frames. We test each building hypothesis according to its consistency with surface orientation estimates in the original frame and the K preceeding frames. The surface orientation estimates are obtained on a per–frame basis using the same method as Lee *et al*.

Enforcing consistency with multiple views improves hypothesis evaluation considerably, particularly since the surface orientation estimates of any one frame are often noisy and incomplete.

6. Concluding Remarks

References

[1] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 44–57, Berlin, Heidelberg, 2008. Springer-Verlag. 2

[2] P. Buschka and A. Saffiotti. A virtual sensor for room detection. In *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on*, volume 1, pages 637–642 vol.1, 2002. 2

[3] M. H. David Changsoo Lee and T. Kanade. Geometric reasoning for single image structure recovery. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009. 1, 2, 5, 6

[4] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Manhattan-world stereo. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1422–1429, 2009. 2

[5] A. Golovinskiy, V. G. Kim, and T. Funkhouser. Shape-based recognition of 3d point clouds in urban environments. In *Proc 12th IEEE Int Conf on Computer Vision*, volume 2, 2009. 2

[6] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *Proc 12th IEEE Int Conf on Computer Vision*, volume 2, 2009. 2

[7] D. Hoiem, A. A. Efros, and M. Hébert. Geometric context from a single image. In *Proc 10th IEEE Int Conf on Computer Vision*, pages 654–661, 2005. 2

[8] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007. 1

[9] J. Koseckà and W. Zhang. Video compass. In *Proc 7th European Conf on Computer Vision*, volume 2353 of *Lec-*

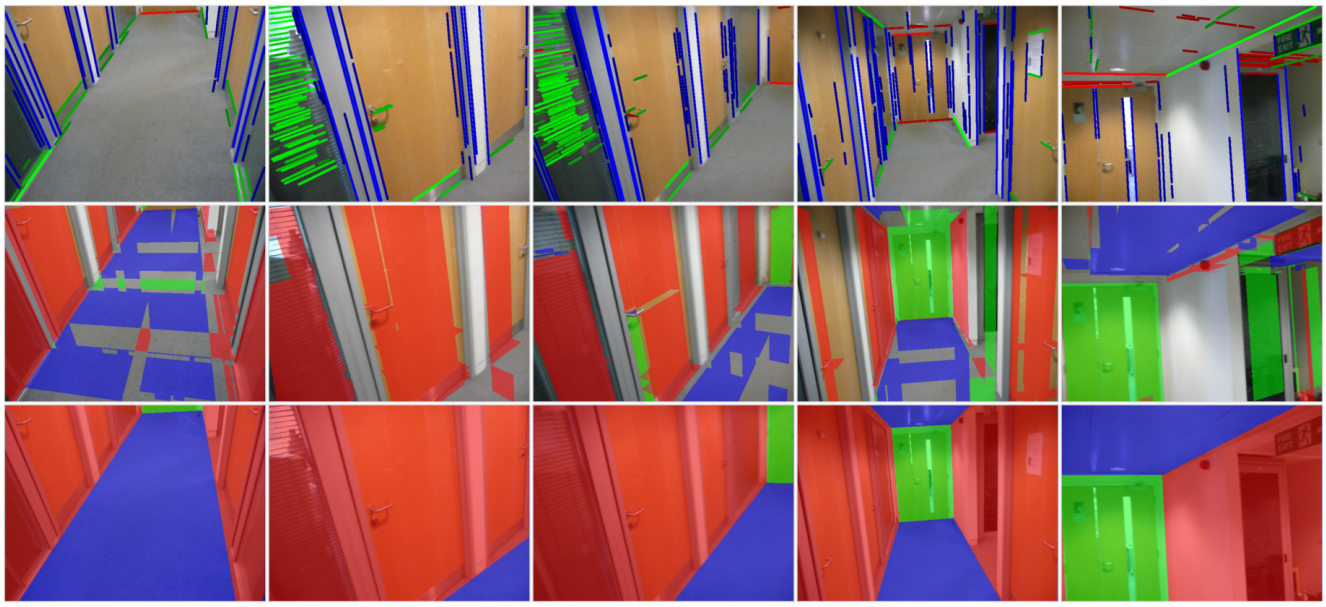

Figure 6. The final reconstruction for the "lab" sequence. The top row shows the line segments identified by the guided line search; the second row shows the surface orientation estimates from the individual frames; the bottom row shows the final model projected into five frames. The orientation estimates shown in the second row are noisy and incomplete but we are able to obtain an accurate model by combining information from all the views. The reconstruction accurately captures the primary surfaces within the scene.

ture Notes in Computer Science, pages 4: 476–490. Springer, 2002. 3, 4, 5

[10] I. Posner, D. Schroeter, and P. Newman. Online generation of scene descriptions in urban environments. *Robot. Auton. Syst.*, 56(11):901–914, 2008. 2

[11] A. Sarti, R. Malladi, and J. A. Sethian. Subjective surfaces: A geometric model for boundary completion. *International Journal of Computer Vision*, 46(3):201–221, 2002. 2

[12] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009. 2

[13] J. Shufelt. Performance evaluation and analysis of vanishing point detection techniques. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(3):282–288, Mar 1999. 2

[14] C. Stachniss, O. Martnez-mozos, A. Rottmann, and W. Burgard. Semantic labeling of places. In *in Proceedings of the International Symposium on Robotics Research*, 2005. 2

[15] J.-P. Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *Proc 12th IEEE Int Conf on Computer Vision*, volume 2, 2009. 3

[16] T. Tuytelaars, M. Proesmans, and L. J. V. Gool. The cascaded hough transform as support for grouping and finding vanishing points and lines. In *AFPAC '97: Proceedings of the International Workshop on Algebraic Frames for the Perception-Action Cycle*, pages 278–289, London, UK, 1997. Springer-Verlag. 2

[17] J. Xiao and L. Quan. Multiple view semantic segmentation for street-view images. In *Proc 12th IEEE Int Conf on Computer Vision*, volume 2, 2009. 2