Segmentation and Tracking of Partial Planar Templates

Abdelsalam Masoud
Colorado School of Mines
Golden, CO 80401
amasoud@mines.edu

William Hoff
Colorado School of Mines
Golden, CO 80401
whoff@mines.edu

Abstract

We present an algorithm that can segment and track partial planar templates, from a sequence of images taken from a moving camera. By "partial planar template", we mean that the template is the projection of a surface patch that is only partially planar; some of the points may correspond to other surfaces. The algorithm segments each image template to identify the pixels that belong to the dominant plane, and determines the three dimensional structure of that plane. We show that our algorithm can track such patches over a larger visual angle, compared to algorithms that assume that patches arise from a single planar surface. The new tracking algorithm is expected to improve the accuracy of visual simultaneous localization and mapping, especially in outdoor natural scenes where planar features are rare.

1. Introduction

For mobile robot applications, it is important to perform Simultaneous Localization and Mapping (SLAM). Visual sensors (*i.e.*, cameras) are attractive for SLAM due to their low cost and low power. Much research has been performed on VSLAM (visual SLAM), and many successful approaches have been demonstrated. Most of these approaches detect feature points in the environment, using an interest point operator (*e.g.*, [1]) that looks for small textured image templates. These templates are then tracked through subsequent images, and their 3D locations, along with the camera motion, are estimated.

In order to track image templates, most existing algorithms assume (either implicitly or explicitly) that each image template is the projection of a single planar surface patch. If the patch is planar, then its appearance can be accurately predicted in subsequent images. For example, a homography (projective) transformation can accurately model the deformation of the image template from the reference image to the current image. Even if a surface is curved, it can appear to be locally planar if the patch size is small enough. However, as the distance between the reference camera and the current camera increases, the prediction error of a curved patch also increases. Tracking will eventually fail when the camera has moved far enough.

A more difficult problem occurs when the template encompasses two disjoint surfaces, which may be widely separated in depth. Unfortunately, such templates often are detected by interest point operators, because the boundary between the surfaces often yields good image texture. However, even small camera motion will cause tracking to fail in such cases.

Some environments, especially outdoor natural environments, have many non-planar surfaces. Figure 1 shows the top 64 points that were automatically detected by an interest point operator [1], using a template window size of 15x15 pixels. By visual inspection, 36 of these templates encompass more than one surface, and 28 templates cover only one surface. Although this analysis is qualitative, it does indicate that outdoor natural scenes can contain many discontinuities. Tracking of these templates will fail after a short time, using tracking algorithms that make the single-plane assumption.

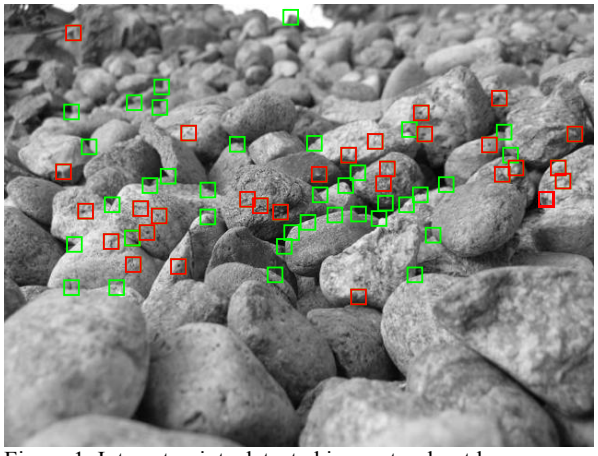


Figure 1 Interest points detected in a natural outdoor scene. Templates that appear to encompass more than one surface are shown as green; the others are shown as red.

If we can model the true 3D structure of a patch, then we can more accurately predict its appearance, and potentially track it over a longer distance. The accuracy of VSLAM can be improved by tracking points over long distances as the camera moves. When a feature is visible over a large visual angle, the error in its estimated 3D location is reduced. This is why wide field of view cameras are preferred for VSLAM [2].

In this work, we present such an algorithm that

estimates the 3D structure of a patch, as it is tracked through a sequence of images. We assume that the image template is the projection of a planar surface, but some of the points in the template may not belong to that surface (*i.e.*, they may belong to other surfaces). We automatically identify the points belonging to the "dominant" plane of the patch, and the parameters of that plane. This allows us to accurately predict the appearance of the pixels belonging to the dominant plane, and ignore the others. As a result, the algorithm is better able to track templates that encompass surface discontinuities.

The novelty of this work is that we can track partial planar templates, over a wider visual angle, as compared to traditional methods that assume that the template consists of a single plane. The new tracking algorithm is expected to improve the accuracy of VSLAM, especially in outdoor natural scenes where planar features are rare. Incorporating the tracking algorithm into VSLAM is beyond the scope of this paper, but will be done as future work.

Note that we make no assumptions about the color distributions of the regions (*e.g.*, that the foreground is lighter than the background). As a result, methods such as mean-shift [3] that segment and track based on color differences are not applicable.

The remainder of this paper is organized as follows: Section 2 presents previous related work in feature tracking. Section 3 gives our overall approach and Section 4 provides a detailed description of the algorithm. Section 5 presents experimental results, and Section 6 provides conclusions.

2. Previous Work

One approach for feature tracking uses learning algorithms to train classifiers that can recognize features undergoing a set of hypothesized deformations [4]. Typically, affine deformations are assumed; thus these algorithms also make the assumption of a single planar patch. If not true, tracking will fail after a short time.

Other related work [5, 6] includes approaches that fit planes in order to perform scene reconstruction. These algorithms try to find large planar regions that can model the scene. These approaches are not applicable to fitting small planar patches.

The most related work to ours includes algorithms that search for a warping image transformation function that registers the reference image patch and the current image patch. The Lucas-Kanade algorithm and its variants [7] is a good example of this. These algorithms compute the image deformation, with parameters $\mathbf{P}$, of the transformation between the reference template $T(\mathbf{x})$ and a region of the current image $I_t(\mathbf{x})$, so as to minimize the sum of squared differences (or a similar residual error measure):

$$E_{SSD} = \sum_{\mathbf{x}} \left[I_t(\mathbf{W}(\mathbf{x};\mathbf{P})) - T(\mathbf{x}) \right]^2 . \quad (1)$$

An affine transformation is often used to model the transformation, but a homography is the most accurate transformation of a planar patch [2].

It is possible to weight each pixel in the template when computing its residual error. Baker, *et al* [7] suggest using a weighting function proportional to the magnitude of the gradient at that pixel; the idea being that those points should be more reliable in the presence of noise. Hager, *et al* [8] compute weights using an iteratively reweighted least squares method, in order to identify points that may be occluded.

We use a similar method, but estimate the reliability of each pixel in terms of how well it fits the dominant plane, over a sequence of images. Specifically, we compute the posterior probability $p(\mathbf{x} \in D | r_{1:t})$ that point $\mathbf{x}$ belongs to the dominant plane D, given residual measurements from times 1..t. We then use this as a weighting function when matching the template to the next image at time t+1, and for re-estimating the parameters of the plane.

3. Overall Approach

We first detect interest points automatically in the first (reference) image. A small square template is extracted around each interest point (in our work, we use size 15x15 pixels). These templates are then tracked to subsequent images using a standard VSLAM approach, where we assume the feature points arise from a single plane. As long as the camera motion is sufficiently small, we can track features in this manner. We also estimate the 3D positions of the features (as well as the camera motion). We call this the "whole plane" method.

After the camera has moved sufficiently far, the image residual error of a template may rise to a level such that we deem the tracking of that template has failed. If this happens, we hypothesize that that the template may consist of more than one surface, and switch to another method to continue tracking this template. This method estimates the 3D parameters of the dominant plane, and the pixels belonging to the dominant plane. We call this the "partial plane" method.

The partial plane method incrementally updates the structure and segmentation of each template, as new images are acquired. Thus, the template can be tracked more reliably in subsequent images, since its appearance can be predicted more accurately. The details are described in the next section.

4. Detailed Description

We represent a plane using the equation $\mathbf{n}^T\mathbf{X} = d$, where $\mathbf{n}$ is the normal vector, d is the perpendicular distance to the origin, and $\mathbf{X}$ is any point on the plane. Given two images of the plane, with rotation $\mathbf{R}$ and

translation **t** between the cameras, corresponding points between the two images are related by $\mathbf{x}_2 = \mathbf{H}\,\mathbf{x}_1$, where **H** is the homography matrix given by:

$$\mathbf{H} = \mathbf{K}\left(\mathbf{R} + \mathbf{t}\,\mathbf{n}^T / d\right)\mathbf{K}^{-1}, \quad (2)$$

and **K** is the camera intrinsic parameter matrix.

4.1. Initialization

When we first begin to track a template using the partial plane method, we perform an initialization step. We need to estimate the depth of the dominant plane, and also the probability that each pixel belongs to the dominant plane. Initially, we assume that the surface normal points towards the camera.

A coarse set of candidate planes is hypothesized, corresponding to a set of depths centered on the current depth estimate (we use ±10 depth values), for a total of 21 candidates. For each candidate plane, we use the hypothesized planar parameters to warp the current image to register it to the template image. The plane that yields the lowest residual error is a candidate for the dominant plane. We then label pixels where the residual error is small, as belonging to this plane.

Now, it is possible that the remaining unlabeled points also belong to a plane. If so, and this second plane is closer to the camera than the first plane, then we identify this as the dominant plane. This is because the closer (foreground) plane may occlude the other as the camera moves, and we expect a more consistent appearance from the foreground portion of the patch.

To find the second plane, we find the candidate plane that has the lowest residual error for the unlabeled points. If this plane is closer than the first one, we choose it to be the dominant plane, as long as it is large enough (we use a value of 20% of the area of the whole template).

Finally, we compute the posterior probability of each pixel belonging to the dominant plane using Bayes' rule:

$$p(\mathbf{x} \in D \,|\, r_1) = p(r_1 \,|\, \mathbf{x} \in D)\, p(\mathbf{x} \in D) / p(r_1) \quad (3)$$

where r_1 is the residual error at pixel **x**, and

$$p(r_1) = p(r_1 \,|\, \mathbf{x} \in D)\, p(\mathbf{x} \in D) + (r_1 \,|\, \mathbf{x} \in \overline{D})\, p(\mathbf{x} \in \overline{D}). \quad (4)$$

Here, the likelihood probability $p(r_1|\mathbf{x} \in D)$ is computed using the normal distribution $N(0, \sigma^2)$, where σ^2 is the expected variance of the pixel values due to noise. $p(r_1|\mathbf{x} \in \overline{D})$ is assumed to be the uniform distribution, since we have no knowledge of the background region. The *a priori* probabilities $p(\mathbf{x} \in D)$ and $p(\mathbf{x} \in \overline{D})$ are assumed to be 0.5 each.

4.2. Refinement of Planar Parameters

As each new image I_t is acquired, we update the planar parameters (**n**, d) of the dominant regions, assuming that the posteriors $p(\mathbf{x} \in D|r_{1:t-1})$ are known, as well as the camera motion.

A non-linear optimization algorithm [9] is used to search for the planar parameters that minimize the sum of the residual errors for all pixels within the template:

$$E = \sum_{\mathbf{x}} \rho(r_t; \sigma) \quad (5)$$

where

$$r_t = \left(I_t\left(\mathbf{W}(\mathbf{x};\mathbf{P})\right) - T(\mathbf{x})\right) \times p(\mathbf{x} \in D \,|\, r_{1:t-1}) \quad (6)$$

is the residual error between the current image and template image at pixel **x**, weighted by the probability that the pixel belongs to the dominant plane. For robustness, the Geman-McClure error function [10] is used:

$$\rho(r;\sigma) = r^2 / (r^2 + \sigma^2) \quad (7)$$

4.3. Refinement of Pixel Probabilities

After the planar parameters have been updated, we next update the posterior probability of each pixel as belonging to the dominant plane, assuming the planar parameters of the region (as well as the camera motion) are known. For simplicity we assume that the residual measurements are independent of each other over time. This is done using the discrete Bayes filter [11]:

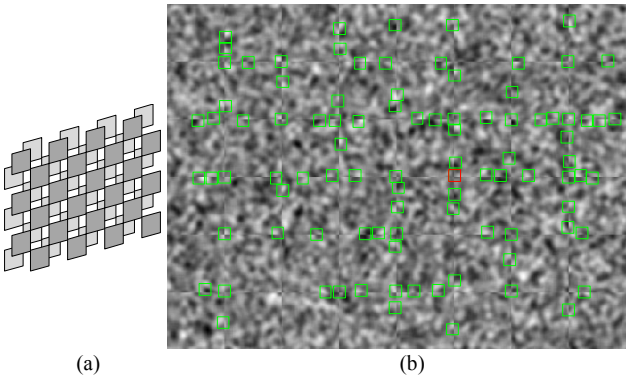
$$\begin{aligned} p(\mathbf{x} \in D \,|\, r_{1:t}) &= \eta\, p(r_t \,|\, \mathbf{x} \in D)\, p(\mathbf{x} \in D \,|\, r_{1:t-1}) \\ p(\mathbf{x} \in \overline{D} \,|\, r_{1:t}) &= \eta\, p(r_t \,|\, \mathbf{x} \in \overline{D})\, p(\mathbf{x} \in \overline{D} \,|\, r_{1:t-1}) \end{aligned} \quad (8)$$

where the value of η is chosen so that the posterior probabilities sum to 1 for each pixel. When updating the probabilities, we use logs to avoid numerical instability.

As before, the likelihood probabilities $p(r_1|\mathbf{x} \in D)$ and $p(r_1|\mathbf{x} \in \overline{D})$ are computed using the normal distribution and the uniform distribution, respectively.

4.4. Example of Synthetic Image

To illustrate our method, we show results on a synthetic image. Two planar surfaces were created in a checker-board pattern (Figure 2a). Both planes were perpendicular to the camera's line of sight. The two surfaces were textured with random noise using the same parameters.



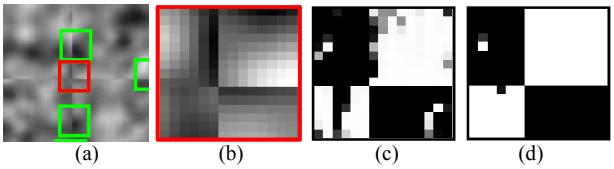
(a) (b)

Figure 2. Synthetic image. (a) 3D structure. (b) Interest points.

Next, interest points were detected in the synthetic

image (Figure 2b), using 15x15 templates. It appears that many of the templates encompass two surfaces.

The camera was then translated to the right for 15 frames. The result of tracking one of the templates (the template outlined in red in Figure 2) is shown in Figure 3. This template encompasses two planar regions, because it is located at one of the corners of the checker-board pattern. The point was tracked first using the "whole plane" method for two images, and then the algorithm switched to the partial plane method on the third image; because the residual error exceeded a threshold (we used a threshold of 15).



(a) (b) (c) (d)

Figure 3. Results on one of the templates in the synthetic image sequence. (a) Location of the template, to provide context. (b) Image template. (c,d) Initial and final probability images.

The initial probability image is shown in Figure 3(c). White indicates high probabilities for $p(\mathbf{x} \in D|r_1)$ for those pixels. This template was then tracked through the rest of the image sequence. The final probability image is in Figure 3(d). The algorithm appeared to correctly segment the template and identify the pixels belonging to the dominant (foreground) plane, with only a few errors.

5. Results on Real Images

The algorithm was applied to real image sequences that have been used to evaluate VSLAM algorithms [12]. The main metric by which we evaluated the algorithm was to measure how long we could track templates using the new method, as compared to the traditional method that used the assumption of a single plane for each template. We also qualitatively examined the segmentation result for each template to see if it appeared to be correct (as we did not have ground truth depths for these scenes, we could not evaluate the segmentation results quantitatively).

In each sequence, we found interest points in the first image, and then tracked their templates through subsequent images. Points were eliminated if they moved outside the field of view, or if the residual error became so large that tracking was assumed to have failed. As a result, the number of tracked templates gradually decreased over time (although in a full VSLAM system implementation, new interest points would be continuously found).

In some cases, points were obviously mismatched, even though the residual error was not large. We manually terminated the tracking of those points. In a full VSLAM implementation, such mismatches could be automatically identified using a method such as RANSAC and fitting to a fundamental matrix. Although this step was out of scope for this paper, we plan to implement it in future work.

The "campus" sequence consisted of 640x480 pixel images, where the camera translated to the right, with a small amount of panning about the vertical axis. Figure 4 shows the initial templates on the first image.

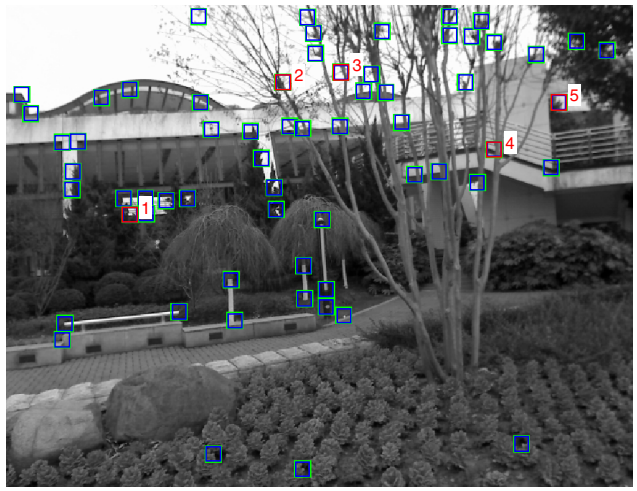


Figure 4. The "campus" image sequence, showing the initial templates (61) on the first (reference) image.

Figure 5 shows the final segmentation results for a few selected templates.

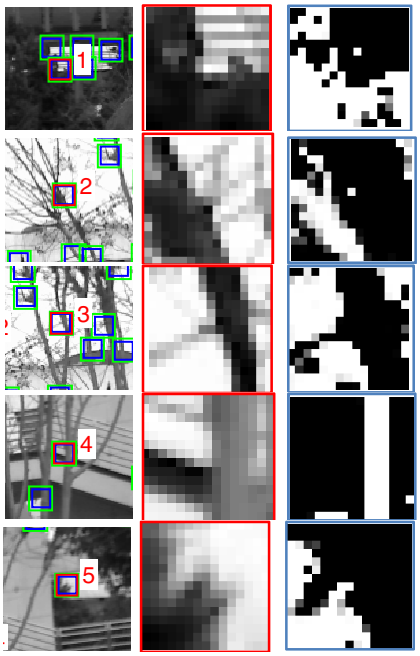


Figure 5. Segmentation results for the "campus" image sequence. Left to right: location of example templates, the image templates, and the final probability image, i.e., $p(\mathbf{x} \in D|r_{1:t})$.

Most appear to be correct, meaning that high probability

values correspond to the dominant (foreground) plane. One of the templates (#3) appears to have an incorrect segmentation, in which the dominant plane corresponds to the background (sky) region, rather than the foreground (tree) region.

Figure 6 shows, for each template, the number of images that that template was tracked in the "campus" sequence. A red bar means that the template was tracked using the "whole plane" method, and a green bar means that it was tracked using the "partial plane" method. Initially, all features are tracked using the "whole plane" method, but when that method fails, some switch to the "partial plane" method.

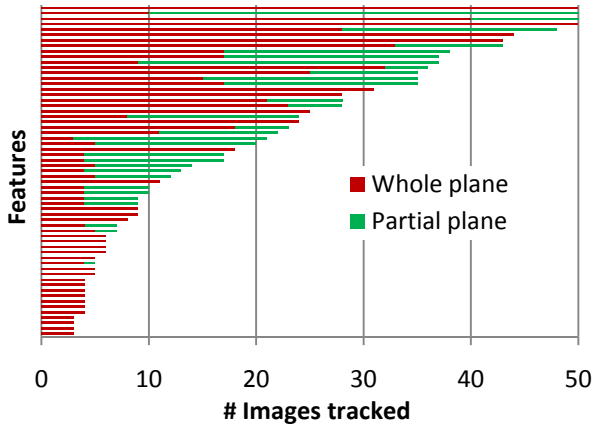


Figure 6. "Campus" sequence. The bars show the number of images tracked for each of the 61 features.

The "flower" image sequence consists of 960x540 pixel images, where the camera translated to the right, moved forward, and panned about the vertical axis. Figure 7 shows the initial templates on the first image.

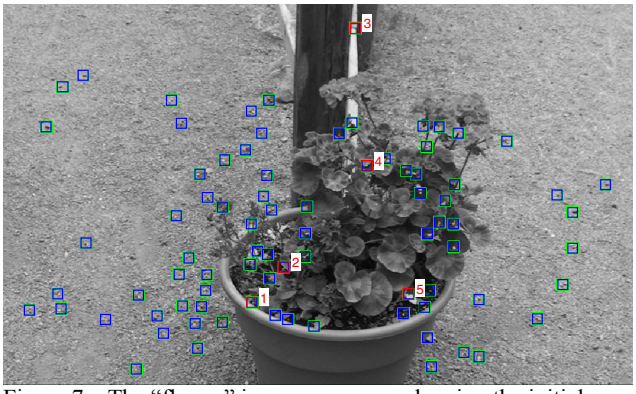


Figure 7. The "flower" image sequence, showing the initial templates (68) on the first (reference) image.

Figure 8 shows the final segmentation results for a few selected templates. Four of these appear to have correct segmentation, but template #4 appears to be tracking the background region.

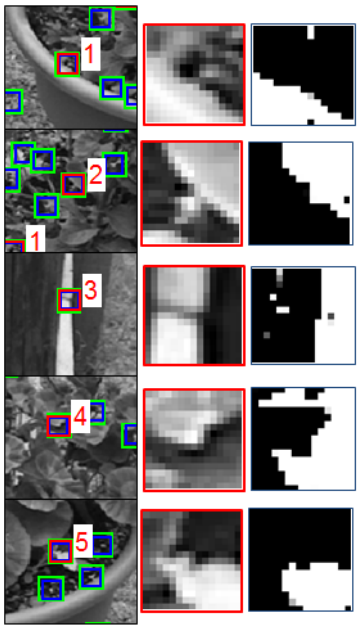


Figure 8. Segmentation results for the "flower" image sequence. Left to right: location of example templates, the image templates, and the final probability image, i.e., $p(\mathbf{x} \in D|r_{1:t})$.

The "road" image sequence consists of 960x540 pixel images, where the primary motion of the camera was a translation to the right. Figure 9 shows the initial templates on the first image.

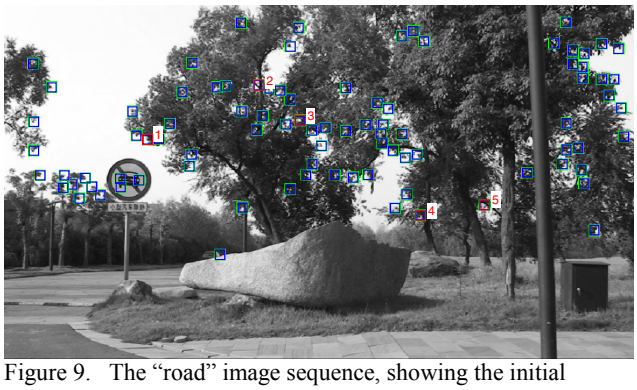


Figure 9. The "road" image sequence, showing the initial templates (68) on the first (reference) image.

Figure 10 shows the final segmentation results for a few selected templates. All of the example templates appear to have correct segmentation, although there is quite a bit of noise.

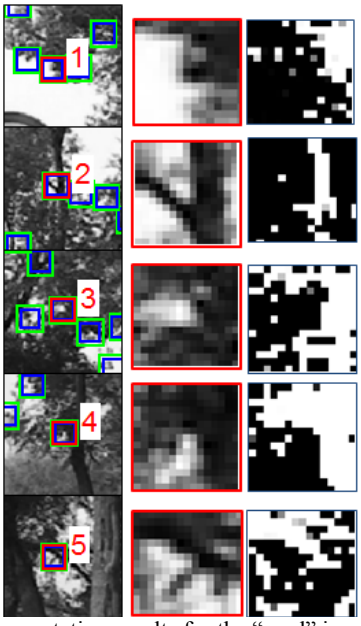


Figure 10. Segmentation results for the "road" image sequence. Left to right: location of example templates, the image templates, and the final probability image, i.e., $p(\mathbf{x} \in D|r_{1:t})$.

Quantitative results for the number of images tracked are shown in Table 1. The table shows that templates can be tracked much longer using both the "whole plane" and "partial plane" methods, as compared to using only the "whole plane" method.

Table 1. Average number of images that templates are tracked, using the "whole plane" method only, versus both "whole" and "partial" plane methods.

Sequence	*"Whole" plane method only*	*Both "whole" and "partial" plane methods*
Campus	13.3	19.2
Flower	13.0	22.6
Road	16.7	21.1

6. Conclusions

We have presented a novel algorithm to segment and track partial planar templates, using a sequence of images from a moving camera. Unlike existing algorithms that assume a feature arises from a single planar patch, our algorithm can handle the case where the patch encompasses more than one surface. Such cases occur often in outdoor natural scenes, where large planar surfaces are rare. We showed that our algorithm can estimate and track such features over a large visual angle, compared to algorithms that assume a patch contains only a single plane. Being able to track features over a larger visual angle should greatly improve the performance of VSLAM. Future work will incorporate our new tracking algorithm into a VSLAM system and measure the improvement in accuracy of motion and structure estimates.

Although we did not specifically focus on run time issues when developing our prototype, we note that the algorithm should be fast when properly implemented. The types of operations performed are typical in real-time VSLAM systems, and the computation of the probability images is very fast.

Finally, although our work was motivated by tracking in outdoor scenes, we note that even man-made indoor environments have problems with non-planar templates. Often interest points are often detected at the boundary between disjoint surfaces, because they produce good image texture. Thus, our algorithm could be beneficial in such scenes as well.

References

[1] J. Shi and C. Tomasi (1994). "Good features to track." CVPR, pp. 593-600, 1994.

[2] A. Davison, *et al*. (2007). "MonoSLAM: Real-time single camera SLAM." *IEEE Trans. on PAMI*, 29(6):1052-1067.

[3] D. Comaniciu and P Meer (2002). "Mean shift: A robust approach toward feature space analysis" *IEEE Trans. on PAMI*, 24(5):603–619.

[4] M. Ozuysal, *et al*. (2010). "Fast keypoint recognition using random ferns". *IEEE Trans on PAMI*, 32(3).

[5] F. Fraundorfer, *et al*. (2006). "Piecewise planar scene reconstruction from sparse correspondences." *Image and Vision Computing*, 24(4):395-406.

[6] D. Fouhey, *et al*. (2010). "Multiple Plane Detection in Image Pairs using J-Linkage." *Int'l Conf on Pattern Recognition*, pp. 336-339.

[7] S. Baker, *et al*. (2004). "Lucas-Kanade 20 Years On: A Unifying Framework". *Int'l J. of Computer Vision*, Vol. 56, pp. 221-255.

[8] G. Hager and P. Belhumeur, "Efficient Region Tracking With Parametric Models of Geometry and Illumination," *IEEE Trans on PAMI*, 20(10), 1998.

[9] J. Lagarias, *et al*. (1998). "Convergence Properties of the Nelder-MeadSimplex Method in Low Dimensions," SIAM Journalof Optimization, 9(1):112-147.

[10] D. Geman and G. Reynolds (1992). "Constrained restoration and the recovery of discontinuities." *IEEE Trans on PAMI*, 14(3):367-383.

[11] S. Thrun, *et al*, "Probabilistic Robotics," MIT Press, 2005.

[12] G. Zhang, *et al*. (2009). "Consistent depth maps recovery from a video sequence," *IEEE Trans on PAMI*, 31(6): 974-988.