Robust Pose Estimation in Untextured Environments for Augmented Reality Applications

Wei Guan* Lu Wang† Jonathan Mooser‡ Suya You§ Ulrich Neumann¶

Computer Graphics and Immersive Technologies Laboratory
University of Southern California

ABSTRACT

We present a robust camera pose estimation approach for stereo images captured in untextured environments. Unlike most of existing registration algorithms which are point-based and make use of intensities of pixels in the neighborhood, our approach imports line segments in registration process. With line segments as primitives, the proposed algorithm is capable to handle untextured images such as scenes captured in man-made environments, as well as the cases when there are large viewpoint changes or illumination changes. Furthermore, since the proposed algorithm is robust to large baseline stereos, there are improvements on the accuracy of 3D points reconstruction. With well-calculated camera pose and object positions in 3D space, we can embed virtual objects into existing scene with higher accuracy for realistic effects. In our experiments, 2D labels are embedded in the 3D scene space to achieve annotation effects as in AR.

Keywords: augmented reality, pose estimation, image registration

1 INTRODUCTION

The estimation of camera pose, which includes camera position and orientation, appears repeatedly in computer vision in many contexts. It is also the fundamental problem in augmented reality, where synthetic objects are inserted into real scenes. Robust and accurate pose recovery is therefore essential in almost all AR applications.

In recent years, researchers have proposed many natural features like SIFT [3], MSER [4], Harris-Affine [2] etc. These features are based on intensities of the neighborhood around some interest point, and such neighborhood is usually of particular shape such as rectangles or ellipses. However, one problem of such point-based descriptors is that they cannot handle the cases when there are large deformations caused by large viewpoint changes, or the cases when there are very few textures. In these aforementioned challenging cases, line-based descriptors [5, 1, 6], are considered to be more reliable, and they are also more tolerant to large illumination changes.

We propose a novel method that can recover camera poses for images captured in untextured environments. A robust image registration algorithm is provided that can handle cases like untextured scenes, large viewpoint changes and complicated illumination changes. The method is based on line signature [6] due to its robustness in different environment conditions. Line segments usually exist in both textured and untextured environments. It is also observed that though large deformations can occur in the cases like large viewpoint change, which often causes failure of intensity-based methods, the topology of line segments are quite stable under such deformations. Besides, lines can be detected consistently under different illumination conditions.

Figure 1: Epipolar lines (left) and augmented annotations (right) for stereo images with large viewpoint change.

2 POSE ESTIMATION ALGORITHM

Pose estimation is a fundamental problem in augmented reality. It is also a crucial part in automated initialization process. In untextured environments, it is difficult for existing trackers to be initialized automatically since most point descriptors are calculated basing on intensities in texture areas. In this section, we propose a robust algorithm that can calculate the camera pose for stereo views in man-made environments where there are few textures.

2.1 Points Correspondence from Line Segments

While point-based descriptors like SIFT can provide pixel-level matchings, line segments can only provide line-level matchings. The segment endpoints cannot offer pixel-level matching since they are often inaccurately detected. In another word, the endpoints of matching segment pair are not exact the matching points.

One solution to this is first grouping the segments that are coplanar in 3D space. Segments that are coplanar will follow the properties of homographs. So this could be done by checking whether one group of segments can be generated through projective transformation of the same group of segments in the other image. If two segments are coplanar, their intersection point should match with the intersection of the corresponding two segments in the other image. Therefore, point-level matching can be achieved. While this is a good way to provide point correspondence, we found that it is usually unnecessary to group the coplanar segments. Since segments on the same surface are often near to each other, we can assume that it is more likely to be coplanar if the segment are close enough to each other and un-coplanar if they are far away. We calculate the intersection points for all the nearby pairs and use RANSAC to

*e-mail: wguan@usc.edu
†e-mail:luwang@usc.edu
‡e-mail:moosergraphics.usc.edu
§e-mail:suyayimsc.usc.edu
¶e-mail:uneumanngraphics.usc.edu

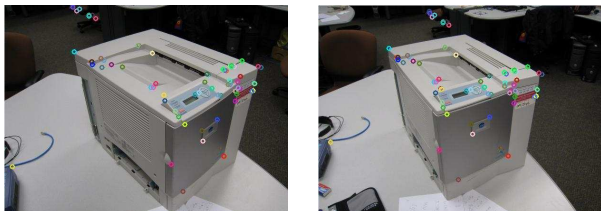

Figure 2: Point correspondences found by segment intersections.

find out the inliers, i.e. the intersections of segments that are truly coplanar.

This procedure can be achieved through the following steps.

Step 1: For all the segments, we remove those that are extremely short. The directions of short segments are sensitive to endpoints detections. The errors in endpoint detections will cause large errors in locations of intersection points. In our experiments, we set the threshold to be 5 pixels length.

Step 2: For each pair of segments (seg_1, seg_2), we calculate the distance between the two mid-points, d_{12}. If $d_{12} > max(l_1, l_2)$, where l_1 and l_2 represent segment lengthes, i.e. the distance is too far relative to the segment length, we ignore this pair and continue with the next pair. If the segments are close to each other, Step 3 is followed.

Step 3: Even though the two segments are close enough, we reject the segment pair that are parallel or nearly parallel (less than 5 degrees), and go back to Step 2 for the next pair. The pair is ignored because the intersection of such nearly-parallel segments is too sensitive to the segment directions. In other words, these intersections are not reliable if the lines are not well fitted.

Step 4: With the pair passing Step 2 and Step 3, the intersection point is calculated. Let P_1 and P_2 be the two endpoints of seg_1, P_1' and P_2' be the two endpoints of seg_2, and Q be the intersection point. With λ and μ as real number, we have,

$$Q = P_1 + \lambda(P_2 - P_1) \tag{1}$$

$$Q = P_1' + \mu(P_2' - P_1') \tag{2}$$

If the intersection point is too far from either of the two segments, it is quite possible that the two segments do not really intersect in the three-dimensional space. However, such intersection points could also be explored as point correspondences. In our experiments, we keep such intersection points only if $\lambda, \mu \in (-1, 2)$, i.e. the distance to a segment is at most the length of the segment.

Step 5: Goto Step 2 for the next pair of segments until all pairs are processed.

Step 6: With all candidate matching points obtained from intersecting segments, we compute the fundamental matrix and use RANSAC to find out the inliers and remove false intersections.

The point correspondences are shown in Fig. 2. We see that some of the points are real line intersections and some are not. However, these "virtual" intersections add more constraints and thus are very useful in the process of pose estimation. Therefore, both "real" and "virtual" intersections are used to recover the pose.

For most segments, the endpoints may not be accurately detected and located. However, if the errors are small, these endpoints can still provide some degree of constraints. Therefore, we would also like to make use of such constraints and consider these endpoints as candidate correspondences.

3 Experimental Results

This section describes the experiments conducted to prove the applicability of the proposed pose estimation algorithm under untextured environments. The images used in the experiments are mostly man-made objects and contain few textures. Furthermore, the view change is large and the objects are close to the camera (relative to their scales). The experimental results show that while traditional intensity-based descriptors cannot handle untextured images well, the proposed algorithm is effective in estimating the camera pose with high accuracy. Moreover, the algorithm is more robust to perspective deformations or large viewpoint changes (Fig. 3).

The reason that point-based features often fail in such environments is that they cannot generate sufficient correspondences. The accuracy of camera pose estimation can only be assured when certain amount of correspondences are provided. With the proposed algorithm, sufficient correspondences can be found and the average reprojection error can be limited within 0.1 pixel.

Figure 3: Left to right: the point correspondences with proposed algorithm, epipolar lines and AR annotations.

4 Conclusion

We propose a new method that estimates the camera pose in untextured environments. The method uses lines that exist in such environments. Lines in different views can be matched according to their relative positions. Due to inaccuracy of edge detection, line segments cannot be explored directly in calculation of camera poses. However, if two lines lie on the same plane in 3D space, the intersections of these two lines can be considered corresponding points. Therefore, we make use of such generated points as the point correspondence and calculate the camera pose. Instead of grouping lines that lie on the same plane, we consider every intersection of two close segments as a candidate match, and use RANSAC algorithm to remove the outliers. Segment endpoints are also included in case there are too few intersections. Experiments have shown that while traditional intensity-based descriptors cannot handle the untextured environments, the proposed algorithm can be explored to estimate the camera pose with high accuracy.

References

[1] H. Bay, V. Ferrari, and L. V. Gool. Wide-baseline stereo matching with line segments. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 329–336, 2005.

[2] T. Lindeberg. Direct estimation of affine image deformation using visual front-end operations with automatic scale selection. In *Proceedings of the International Conference on Computer Vision*, pages 134–141, 1995.

[3] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60):91–110, Jan 2004.

[4] J. Matas, S. Obdrzalek, and O. Chum. Robust wide-baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 384–393, 2002.

[5] C. Schmid and A. Zisserman. Automatic line matching across views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 666–671, 1997.

[6] L. Wang, U. Neumann, and S. You. Wide-baseline image matching using line signatures. In *Proceedings of the International Conference on Computer Vision*, 2009.