Iterative reconstruction of large scenes using heterogeneous feature tracking

Rohith MV*, Stephen Rhein*, Guoyu Lu*, Scott Sorensen*,
Andrew R. Mahoney†, Hajo Eicken†, G. Carleton Ray‡, Chandra Kambhamettu*

*Video/Image Modeling and Synthesis (VIMS) Lab,
Dept. of Computer and Information Sciences, University of Delaware, Delaware, USA.

† Geophysical Institute, University of Alaska Fairbanks, Fairbanks, Alaska, USA.

‡ Department of Environmental Sciences, University of Virginia, Charlottesville, Viriginia, USA.

Abstract

With image capturing technology growing ubiquitous in consumer products and scientific studies, there is a corresponding growth in the applications that utilize scene structure for deriving information. This trend has also been reflected in the plethora of recent studies on reconstruction using robust structure from motion, bundle adjustment, and related techniques. Most of these studies, however, have concentrated on unstructured collections of images. In this paper, we propose a feature tracking and reconstruction framework for structured image collections using heterogenous features. This is motivated by the observation that images contain a small number of features that are fast/easy to track and a large number of features that are difficult/slow to track. By tracking these separately, we show that we can not only improve the tracking speed, but also improve the tracking accuracy by using a camera geometry based descriptor. We demonstrate this on a new challenging dataset which contains images of Arctic sea ice. The reconstruction pipeline constructed using the proposed method provides near real time reconstruction of the scene, enabling the user to parse vast amounts of data rapidly. Quantitative comparisons with baseline SFM techniques show that reconstruction accuracy does not suffer.

1. Introduction

As the cost and effort required in capturing images grows smaller, areas where Computer vision techniques, and in specific scene reconstruction algorithms, may be applied grow wider. As structure from motion algorithms are applied to large datasets originating from vehicle mounted cameras, surveillance networks, and other multi-camera/video sources, feature tracking becomes a bottleneck that decides the performance in terms of accuracy and speed. In this paper, we propose a feature tracking and reconstruction framework for structured image collections using heterogenous features. Our approach is motivated by the following observations:

- Feature tracking through exhaustive search is computationally very intensive
- There are different kinds of features in images (Corners, points in homogeneous regions etc)
- Some features are easy/fast to track. But there are only few of such features in each image (E.g., Corners using LKT tracker)
- Some features are difficult/slow to track. There is a large number of such features (Points in planar regions using SIFT)
- If some correspondences are found between two images, they can be used in solving correspondence problem for other points faster.

The novelty of the proposed method is as follows - (a) we track heterogenous set of features in image sequences, (b) we develop a descriptor that incorporates epipolar geometry and feature arrangement information, into a vector that can be matched using L_2 distance, (c) we use the advantages of the various features in an incremental SFM to improve accuracy.

In Section 2, we discuss some of the previous methods in this area. Section 3 provides an overview of our method. Section 4 discusses the various features used in this work and the descriptors used to track them. In Section 5, we present results on a challenging dataset of Arctic sea ice collected aboard the icebreaker RV Polarstern during the cruise ARK-27/3. As the dataset was collected using stereo camera system, the 3D reconstruction from stereo provides an independent basis of evaluation. We provide quantitative evaluation of both tracking and 3D reconstruction errors to show that the scheme of classification of features and the proposed descriptors indeed improve the performance. We provide quantitative evaluation of both tracking and 3D reconstruction errors to show that the scheme of classification of features and the proposed descriptors indeed improve performance.

2. Background

Research related to structure from motion and feature tracking forms a vast component of computer vision literature and hence we restrict ourselves to a discussion of works that are most relevant to the proposed approach. We will discuss only rigid structure from motion methods as most of the data we consider here consists of stationary scenes and the proposed technique is geared towards large scale reconstruction where motion may either be ignored or filtered out. For feature tracking methods, we only concentrate on methods that perform sparse feature matching as these form the input to most of the structure from motion algorithms. Although a denser reconstruction may be obtained from dense wide baseline stereo techniques, the levels of error precludes them from being used in estimation of camera pose.

2.1. Structure from motion

Since the seminal work by Tomasi and Kanade [16], there have been numerous attempts at designing methods that are more accurate, robust to outliers, and scaleable. The method of scaling image coordinates to obtain better reconstruction proposed by Sturm and Triggs [14] was extended by Oliensis and Hartley [10] using an iterative approach. Bundle adjustment [17] provided a method for improving results of structure from motion by exploiting the sparsity in the minimization of the non-linear reprojection error based objective function. An excellent collection of various structure from motion algorithms is available in [11]. There are also methods that perform incremental SFM [12] and SFM of ambiguous image sequences [8]. Unstructured data obtained from internet databases have been used to reconstruct cities [6, 1]. Frahm et al. [6] use cloud based computing to reconstruct the city of Rome in a day whereas Agarwal et al. [1] perform similar reconstructions by using GPU acceleration. Although these methods use large scale databases, the datasets they use are seldom structured. In this paper, we consider the methods that can exploit the structure, in terms of contiguity of image sequences and overlap information, to increase the speed and accuracy of the reconstruction.

2.2. Feature matching

For sake of this discussion, we classify feature matching methods into appearance based and graph based methods. Appearance based methods use description of the image region around a detected feature point to establish correspondence, whereas graph based methods utilize the arrangement and location of feature points. Several detector/descriptors combinations may be employed depending on the nature of the images. SIFT [9], SURF [3] and the more recent FREAK descriptors [2] are some of the descriptors that are shown to be invariant to rotation and affine scaling of the images. The matching process involves building a table of descriptors for each image and solving for correspondence using exhaustive pair-wise comparison of descriptors, or nearest-neighbor based methods. Tensor based node descriptors [4, 5], and adaptive geometric templates [18] are some examples of graph based matching. These methods exploit the invariance in local arrangement of features to establish correspondences. The problem is often formulated as a sub-graph matching problem where the nodes of the graph correspond to the detected features. Most of the tracking methods suffer from two drawbacks - they treat all the features uniformly, they only incorporate epipolar constraints as a post-processing filter. In the proposed method, we design a framework for handling heterogenous features by exploiting their relative strengths.

3. Overview

Figure 1 provides an overview of the proposed method. There are two phases of the algorithm - feature tracking and iterative reconstruction. To reflect the ease with which features may be tracked, they are divided into corners and non-corner features. Corners represent points which are easy to detect, and can be tracked using methods using linear motion models such as Lucas-Kanade-Tomasi (LKT) Trackers [15]. Correspondences are obtained accurately and quickly (pyramid-based trackers which can track several hundred features at frame rate are available in CPU and GPU implementations). Corners that exist in overlapping regions of the image sequence are used for iterative pose estimation. We use 6-point RANSAC algorithm for pose estimation of an orthographic camera provided in [7]. Non-corner features are tracked through a hybrid graph-epipolar based descriptor which uses the corners as reference. Since the error rate in these features is larger, we use them only in triangulation from the camera poses estimated using corners.

4. Tracking corners

Since we want corner features to be quickly and accurately trackable, we use the detector/tracker combination of Shi-Tomasi features [13] with LKT tracker. Although multi-scale LKT trackers can handle large movements between images, owing to the challenging nature of our dataset, these methods cannot satisfactorily track features that are closer to the camera (displacements of about one third of image width). For this we propose a method that approximately aligns the images to reduce the displacement between images. The method is shown in Figure 2. If I_1 and I_2 are two images in a sequence, then four divisions are created horizontally (Figure 2a), since our camera mo-

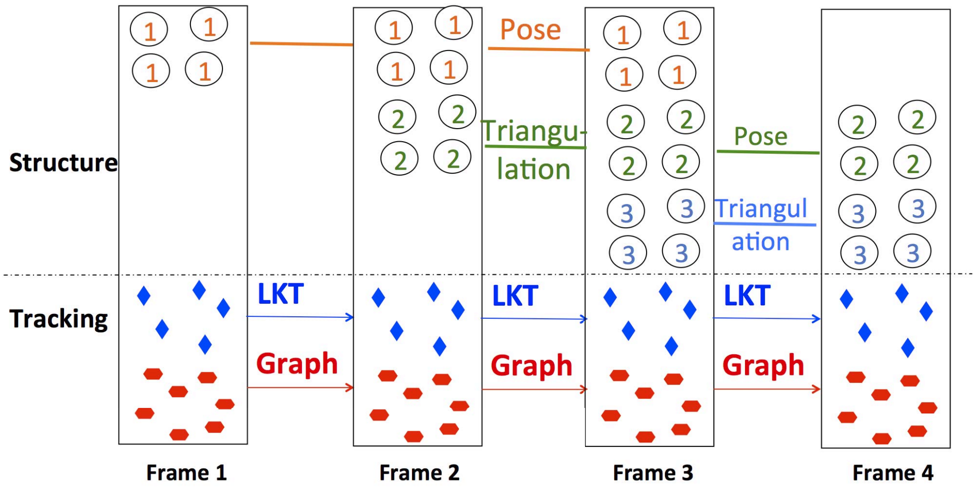

Figure 1. Outline of approach. Two phases of structure estimation and tracking are shown. The features are indicated as circles in first phase and shapes in the second phase. The numbers in the circles indicate the frame at which a feature was first tracked.

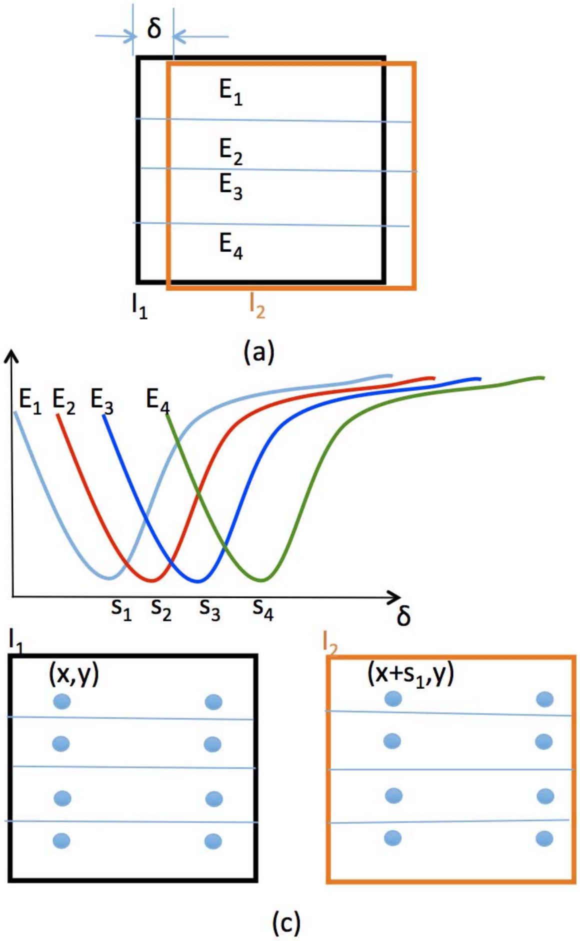

Figure 2. Warping for LKT. (a) Divisions in the images, (b) Block SSD error of each division with respect to shift, and (c) Correspondences based on block matching.

tion is mostly translational along the horizontal axes. The orientation and layout of these divisions are based on the expected camera motion and can be modified accordingly. Approximate movement of each block is calculated via SSD matching between the two images with horizontal shifts (curves shown in Figure 2b). Artificial correspondences are created between the two images by creating points in each division with the corresponding estimated shift (Figure 2c). These correspondences are used to estimate a perspective transform that aligns the two images. When LKT is performed on this warped version of I_2, more points are tracked correctly (Table 1).

5. Tracking Non-corner features

Non-corner features consist of points on edges, flat regions, and other areas where a linear motion model would have ambiguity in establishing correspondences. Though appearance descriptors such as SIFT provide a robust method of matching such features, they are associated with a large computational complexity owing to their large dimensionality (commonly used SIFT is 128 dimensional). The extraction and matching of such large dimensional entities for every frame of a video sequence is time consuming when image sizes are large.

We propose a new feature descriptor which does not depend on appearance of the patch, but only epipolar geometry of camera poses and neighboring feature points. Since the descriptor should be easily matchable using Approximate nearest neighbor (ANN) or clustering methods, we cannot use bi-linear forms such as the Fundamental equation constraint directly. Also, since we have already established correspondences for corner features, we can use them in constraining the match for non-corner features (unlike other graph-based correspondence methods that treat all the features uniformly). The descriptor we propose will have the form $[x, y, d_1, d_2, \alpha_1, \cdots, \alpha_K]$. x, y refers to the position of the feature point. d_1, d_2 are the scalars obtained from epipolar constraints. $\alpha_1, \cdots, \alpha_K$ are from neighborhood constraints. The descriptors are constructed for pairs of images and the correspondences are carried over to obtain tracks.

5.1. Epipolar descriptor

If p_l and p_r are corresponding points in images I_1 and I_2, then they satisfy the following relation:

$$p_l F p_r = 0.$$

F is the fundamental matrix that contains the description of the epipolar geometry. To derive a descriptor embodying the above constraint that can be matched using L_2 norm, a descriptor is defined as seen in Figure 3(a). For obtaining descriptor of a point (x, y) in I_1, we consider two lines, the line joining (x, y) to the first epipole e_1 in I_1 and the line obtained by transforming the point (x, y) using F in I_2. Due to epipolar constraints all the possible correspondences of (x, y) lie on the line indicated in I_2. The feasibility of any point in I_2 being correspondence of (x, y) can be estimated by distance from this line. However, since we cannot encode this as a L_2 distance based descriptor, we use the intercept of the line on an arbitrary vertical line (left edge of image in the case shown). Similarly, we can extract descriptors for points in I_2 by transposing F and using e_2.

5.2. Corner-neighborhood descriptor

The neighborhood descriptor uses corner correspondences to constrain matching of non-corner features. Since the scene is rigid, the arrangement of feature points in the images seldom change. If there are no points of reference, then the arrangement itself can be used as a descriptor (as in [5]). However, since we have some known correspondences, they can be used to provide a reference around which other points move. For every non-corner feature detected, we find the K nearest corners around it ($K = 3$ case shown in Figure 3(b)). α_1 is the orientation of the line connecting the non-corner feature with its nearest corner, α_2 is for the next nearest and so on. The descriptor is made up of these orientations - $\alpha_1, \cdots, \alpha_K$. The estimation of nearest neighbors can be accelerated using tesselations such as Delaunay triangulations, or data structures such as Quadtrees. Since the number of corner points are small compared to non-corners, we tesselate the image with corners as nodes.

6. Results

We test our algorithms on a challenging dataset of Arctic icescapes (regions covered in snow and ice) collected using a stereo camera system. The dataset consists of many swaths hundreds of kilometers in length totaling more than 2500 kilometers along the ARK-27/3 cruise track[1]. The images are 5 megapixels in size and the cameras are calibrated for intrinsic parameters. The data contains a total of about 1,223,355 images with sections of contiguous sequences that have 50,000 or more images. The application demands a sparse reconstruction of the entire region. If the sparse reconstruction can be performed at speeds at which a user can interact, it would facilitate identification of areas where a denser reconstruction is desirable. The images contain little color and large homogeneous regions.

To test the algorithm, we generated 3D models of two regions (Scene 1 and Scene 2) from stereo pairs. These point clouds were then rendered with the original colors to create image sequences (of length 8) of the scene with known correspondences. The 3D model and two contiguous frames from Scene 1 are shown in Figure 4 (a) and (b) respectively.

6.1. Evaluation of tracking

We evaluated the various descriptors proposed in this paper. The correct correspondences between every pair was identified using the ground truth. The average of features found and tracked correctly using the various methods are summarized in Table 1. For corner features, the LKT algorithm with warped and unwarped images show that warping increases correct correspondences by about 75%. We consider SIFT detector/SIFT descriptor as reference for non-corner features. We then evaluate a host of other descriptors against it. These are coded as - XY for only using image coordinate as descriptor, XYE for image coordinate and epipolar descriptor and XYEG for including the neighborhood descriptor. It can be seen that adding epipolar and neighborhood descriptor increases the tracking accuracy significantly. Varying the number of neighbors used does not seem to have a significant impact if $K > 3$. To verify that the matching is truly invariant to appearance of the patch around the feature, we used Harris detector with our descriptor. As expected, the fraction of correct correspondences does not decrease by much. This can be extended to simpler features such as LoG maximas with distance suppression.

6.2. Evaluation of 3D reconstruction

We evaluated the 3D reconstruction produced by the tracks from XYEG matching ($K = 5$) using three different techniques. The average errors are provided in Table 2. Column SFM contains errors from algorithm in which SFM [10] was performed for each set of overlapping features and the results were aligned. This is treated as

[1] Boetius, A; ARK-XXVII/3 Shipboard Scientific Party (2013): Track of POLARSTERN cruise ARK-XXVII/3 (IceArc). Alfred Wegener Institute, Helmholtz Center for Polar and Marine Research, Bremerhaven, doi:10.1594/PANGAEA.796209

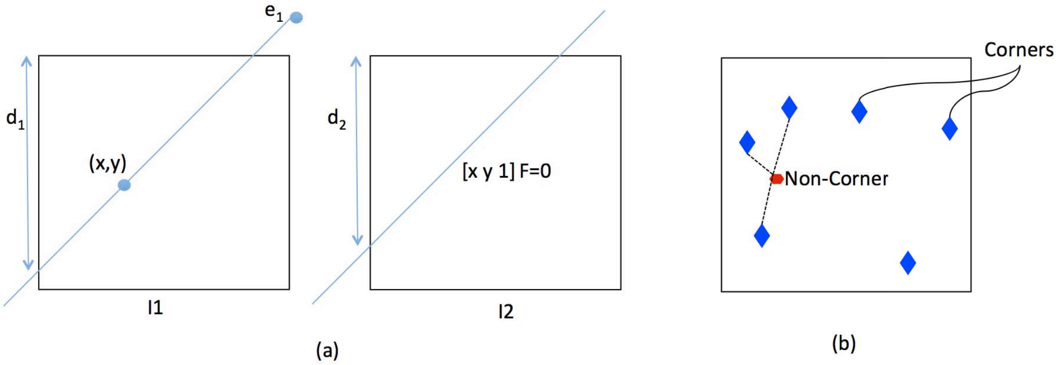

Figure 3. Descriptors for non-corner features. (a) Epipolar feature - consists of two scalars d_1 and d_2 for matching features between images I_1 and I_2. F is the fundamental matrix and e_1 is the first epipole for the image pair. (b) Graph feature - A non-corner point and its neighboring corners are shown. The feature consists of scalars defining the orientations of the lines connecting the non-corner to its neighboring corners.

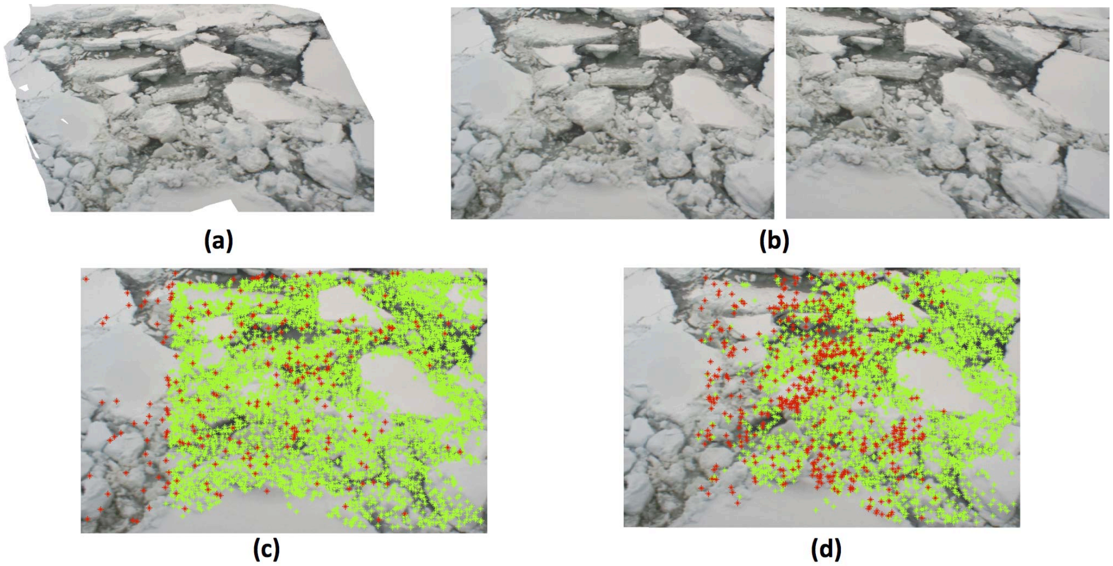

Figure 4. Results of tracking. (a) 3D model of Scene 1, (b) Two contiguous frames in the image sequence, (c) Result of using SIFT detector/SIFT descriptor for matching, and (d) Result of using SIFT detector and XYEG descriptor for matching. In (c) and (d), the correctly tracked points are marked in green and incorrectly tracked points are marked in red.

the reference, as pose for each frame is estimated directly by SFM algorithm. In the second column ISFM(all), SFM was performed on first four frames and poses for subsequent frames and the 3D coordinates of features seen in them were obtained using triangulation. This is the naive version of the incremental SFM algorithm where all features are treated uniformly. As seen in the table, the errors increase due to inaccuracies in tracking the non-corner points. For the third column ISFM(corners), we use the same method as second column, except that only corner points are used for pose estimation. The non-corner points are only triangulated using these estimated poses. This decreases the error significantly compared to ISFM(all) without the need for multiple SFM operations. We wish to highlight that the same feature tracks were used for all the algorithms.

	SFM	ISFM (all)	ISFM (corners)
Scene 1	10.34	60.44	16.44
Scene 2	12.95	72.23	17.19

Table 2. Evaluation of reconstruction error. Average of errors across all points are provided as a percentage of the mean distance of the scene from camera.

7. Conclusions

In reconstructing large scenes, feature tracking is a bottleneck in performance. We observed that images contain a small number of features that are fast/easy to track and a large number of features that are difficult/slow to track. By tracking these separately, we showed that we can not only improve the tracking speed, but also improve the tracking

Detector	Tracker/Descriptor	Features found	Correct Correspondences
Shi-Tomasi	LKT (unwarped)	1000	560
Shi-Tomasi	LKT (warped)	1000	980
SIFT	SIFT	4664	4323
SIFT	XY	4664	2397
SIFT	XYE	4664	2763
SIFT	XYEG (K=1)	4664	3287
SIFT	XYEG (K=2)	4664	3359
SIFT	XYEG (K=3)	4664	3787
SIFT	XYEG (K=5)	4664	3890
Harris	XYEG (K=5)	7838	6540

Table 1. Performance of various tracking schemes. The numbers present averages over two test sequences.

accuracy by using a camera geometry based descriptor. By using a combination of epipolar and neighborhood descriptors, we showed that features can be matched irrespective of their appearance. We demonstrated this on a new challenging dataset which contains images of Arctic sea ice. Quantitative comparisons with baseline SFM techniques showed that reconstruction accuracy did not suffer. In future, we wish to extend this method to include descriptors from multiple images (such as trifocal tensors) and also improve the tracking accuracy by incorporating low dimensional appearance descriptors.

8. Acknowledgements

We would like to thank the RV Polarstern ARK-27/3 crew and scientific party. This work was made possible by NSF CDI-Type I grant 1124664.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building rome in a day. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 72–79, 2009.

[2] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *CVPR*, pages 510–517, 2012.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008. ¡ce:title¿Similarity Matching in Computer Vision and Multimedia¡/ce:title¿.

[4] Z. Cheng, Y. Chen, R. Martin, Y. Lai, and A. Wang. Supermatching: Feature matching using supersymmetric geometric constraints, 2013.

[5] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1980–1987, 2009.

[6] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 368–381, Berlin, Heidelberg, 2010. Springer-Verlag.

[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[8] N. Jiang, P. Tan, and L.-F. Cheong. Seeing double without confusion: Structure-from-motion in highly ambiguous scenes. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1458–1465, 2012.

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, November 2004.

[10] J. Oliensis and R. Hartley. Iterative extensions of the Sturm/Triggs algorithm: Convergence and nonconvergence. *PAMI*, 29(12):2217–2233, 2007.

[11] V. Rabaud. Vincent's SFM Toolbox. `http://vision.ucsd.edu/~vrabaud/toolbox/`.

[12] A. L. Rodriguez, P. E. Lopez-de Teruel, and A. Ruiz. Reduced epipolar cost for accelerated incremental sfm. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 3097–3104, Washington, DC, USA, 2011. IEEE Computer Society.

[13] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600, 1994.

[14] P. F. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proceedings of the 4th European Conference on Computer Vision-Volume II - Volume II*, ECCV '96, pages 709–720, London, UK, UK, 1996. Springer-Verlag.

[15] C. Tomasi and T. Kanade. Detection and tracking of point features. *International Journal of Computer Vision*, 1991.

[16] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.

[17] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment a modern synthesis. In *Vision Algorithms: Theory and Practice, LNCS*, pages 298–375. Springer Verlag, 2000.

[18] H. Veeraraghavan, P. Schrater, and N. Papanikolopoulos. Adaptive geometric templates for feature matching. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 3393–3398, 2006.