Structure from Motion via a Two-Stage Pipeline of Extended Kalman Filters

Brian Clipp
bclipp@cs.unc.edu

Gregory Welch
welch@cs.unc.edu

Jan-Michael Frahm
jmf@cs.unc.edu

Marc Pollefeys
marc@cs.unc.edu

Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC USA

Abstract

We introduce a novel approach to on-line structure from motion, using a pipelined pair of extended Kalman filters to improve accuracy with a minimal increase in computational cost. The two filters, a *leading* and a *following* filter, run concurrently on the same measurements in a synchronized producer-consumer fashion, but offset from each other in time. The leading filter estimates structure and motion using all of the available measurements from an optical flow based 2D tracker, passing the best 3D feature estimates, covariances, and associated measurements to the following filter, which runs several steps behind. This pipelined arrangement introduces a degree of non-causal behavior, effectively giving the following filter the benefit of decisions and estimates made several steps ahead. This means that the following filter works with only the best features, and can begin full 3D estimation from the very start of the respective 2D tracks. We demonstrate a reduction of more than 50% in mean reprojection errors using this approach on real data.

1 Introduction

Structure from motion (SfM) is a well studied problem in computer vision. Most approaches begin with a set of salient 2D image features that are tracked from frame to frame using optical flow or wide baseline feature matching. Feature selection, determining which features to use in the structure from motion, is critical to the accuracy of results. Common approaches include RANSAC [8], robust regression [11] and filtering approaches which use a camera motion model to determine outliers in systems using Kalman or particle filter based 3D trackers [6, 7].

Our novel approach combines two extended Kalman filters that run concurrently on the same measurements in a synchronized producer-consumer fashion, but offset from each other in time. The leading filter generates initial estimates of sparse scene structure and the camera motion by identifying 2D tracks called *inliers* in the total set of 2D feature tracks. The subset of inliers determined by the leading filter provide better information about the camera pose. The leading filter passes their 3D estimates and covariances

BMVC 2007 doi:10.5244/C.21.75

(which have been improved by the influence of many measurements) to the following filter, which operates only on these good feature tracks with reliable initial 3D estimates.

In the experimental evaluation we demonstrate our pipelined approach on real data where it reduces the reprojection errors of the estimated 3D points in the following filter by more than 50%. This reduction in reprojection error reflects the fact that the pipelined two-filter approach only employs measurements that have been found to be consistent with the camera motion in the immediate future. While the improved estimates are delayed in time compared to the newest frame, which might be a concern for on-line applications, the approach allows the user to trade off this delay for improved performance.

In contrast to our approach which uses all temporal correspondence information over multiple frames (chains of matches), typical previous SfM approaches only employ correspondences from a single pair of frames. This is a result of the correlation of their computational cost with the probability of correct correspondences. As the probability of a chain of correspondences is significantly lower than for a single correspondence, previous approaches are often not efficient on chains of correspondences. (For a more complete overview of robust estimation in computer vision we suggest [10].) Our approach is efficient in that a naive approach to looking ahead w frames for inliers would run with $O(wh)$ complexity where h is the cost of one complete structure from motion estimation over all of the frames, while our two-stage filtering approach requires only $O(h)$ time.

In the next section we will discuss work related to the pipelined filter. Section 3 describes the pipelined filter architecture in detail and section 4 presents some experimental results that demonstrate the improvement in reprojection errors by our two-stage (leading-following) pipelined multi-filter approach.

2 Related Work

A key component of any structure from motion system is the estimation of the camera motion in 3D space from 2D feature tracks. Typically the obtained tracks contain a fair number of outliers. Hence the estimator has to simultaneously estimate the camera motion and to classify the tracks into outliers and inliers. Robust estimators are successfully applied to solve this problem in many computer vision applications. The most common technique to deal with outliers is the RANSAC algorithm [8, 20]. It solves the two problems of computing a relation that best fits the data and classifying the data as inliers (correct matches) and outliers. The classification is done by employing a cost function together with a threshold which depends on the expected measurement noise. The relation is then selected as the one with the highest number of inliers or the largest robust likelihood [8]. An inlier with respect to an error function has an error less than a threshold.

When the expected noise is not known beforehand it is difficult to determine the appropriate threshold. Then often robust regression methods are used to estimate the relation of the images and the classification of the data into inliers and outliers [11]. These methods achieve the greatest success when the data belong to a single signal corrupted with random outliers. Miller and Stuart [12] extended the MINPRAN robust regression method [18] to account for data that belong to multiple signals. The MINPRAN operator [18] tolerates a large number of outliers and identifies regions composed completely of outliers.

Tang et al. [19] proposed a tensor voting based approach that poses the problem of estimating the epipolar geometry (the focus of the paper which can be extended to many other estimation problems) as one of finding the most salient hyperplane in a multi-dimensional space. Another popular technique is the Least Median of Squares (LMS) estimation [16]. LMS has been very successful when applied to a lone signal corrupted with outliers but fails completely if the outlier rate is higher than 50%. LMS searches a space of hypothesized fits using an objective function based on the median squared residual.

Another class of estimators adds a camera motion model to assist in detecting outliers. This measurement selection approach is based on a smooth motion model and consensus and is used with a Kalman filter in [1, 3] and a real-time particle filter in [7]. Davison presents a real time extended Kalman filter based visual simultaneous localization and mapping (SLAM) system in [6]. He uses a top down approach to measurement selection, searching for 2D features only in the region they are expected to be in the image based on estimation uncertainty, to minimize computational cost per frame.

Finally, we note that ideas for *fixed point* and *fixed lag* smoothing within a single Kalman filter were introduced by Rauch et al. [13, 14, 15]. The basic idea is to recursively estimate the state at some past time, either at some particular point in time, or following the current time with a fixed delay, using all of the available measurements. By separating the estimation into two pipelined filters we are able to prevent outliers from negatively affecting the second (following) filter, while simultaneously providing the following filter with non-causal initial estimates of the 3D points and covariances. In effect, we obtain some of the benefits of fixed-lag smoothing, but using only the inliers.

3 Two-Stage Measurement Selection and Estimation

The two-stage measurement selection/initialization and final estimation 3D pose filter is composed of two individual extended Kalman filters (EKF). We refer to them as the *leading* and *following* filters. The two filters run concurrently on the same measurements (images) in a synchronized producer-consumer fashion, but offset from each other in time. They are identical except in the way that they initialize 3D feature estimates, where the leading filter initializes points by triangulation and the following filter receives its initial feature position and covariance estimates from the leading filter.

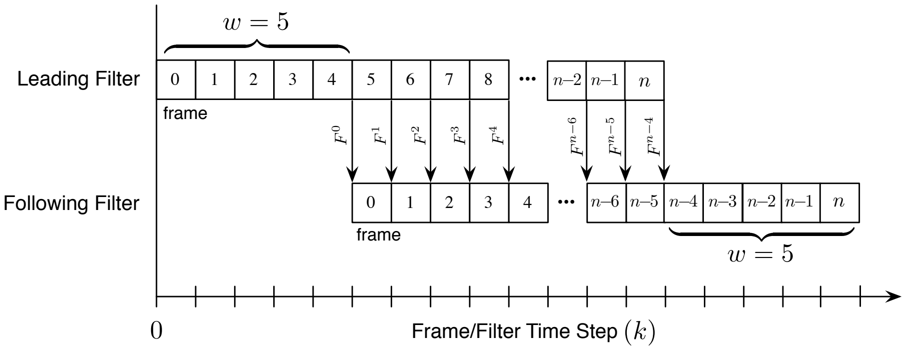


Figure 1: Timing of pipelined leading and following filters for time offset $w = 5$.

Figure 1 shows an example of pipelined leading and following filters for time offset $w = 5$. Once the initial w frames have been processed (the pipeline primed) then at each filter time step k the leading filter passes its latest feature set F^{k-w} for frame $k-w$ to the following filter. Omitting the w for clarity, the feature set F^k for frame k is defined as

$$F^k = \{(x_1^k, \hat{X}_1^k, \Sigma_1^k), \ldots, (x_n^k, \hat{X}_n^k, \Sigma_n^k)\} \tag{1}$$

where x_i^k is the actual measurement (2D projection) of a feature, $\hat{X}_i^k$ is the *estimated* 3D position that feature at time k, Σ_i^k is the corresponding 3D covariance of the estimate, and n is the total number of features. The leading filter spends w time steps attempting to estimate 3D feature locations for frame $k-w$, selecting only the best ones to pass on to the following filter. In the remainder of this section we will first describe a single filter and then describe how the two filters are combined to form the estimation system.

3.1 Individual Filter

While we use an extended Kalman filter for this work, we believe the pipelined approach could be employed with any on-line 3D filters or other estimators. Our filters fuse measurements from a 2D KLT tracker [9, 17], which is an optical flow based 2D tracker that measures the motion of salient features from one frame to the next in a video sequence. The filter's process model uses a smooth motion model for the change in camera position and orientation. It uses a first order Taylor series approximation to relate the state at time k to time $k+1$. This model assumes that the velocity is constant. The estimated 3D features must be static with respect to the world frame to be included in the filter state, and so they are modeled as having zero velocity.

The filter state S^k at time k is made up of the camera's position C^k, orientation θ^k, velocity $\dot{C}^k$, orientation rate $\dot{\theta}^k$ (rotational velocity) and estimates of the 3D position of each of the n features being tracked $X_1^k ... X_n^k$. The filter state is shown in Equation (2),

$$S^k = \left[\begin{array}{ccccccc} C^k & \dot{C}^k & \theta^k & \dot{\theta}^k & X_1^k & \ldots & X_n^k \end{array} \right]^T \tag{2}$$

where again, n is the number of tracked features. The filter's predicted measurement equation is simply the projection of each estimated 3D feature i into the camera at time k given calibration K. In Equation (3) R is the rotation matrix composed from the Euler angle representation of the camera orientation Θ^k.

$$\hat{x}_i^k = K \left[\begin{array}{cc} R^{T^k} & -R^{T^k} C^k \end{array} \right] X_i^k \tag{3}$$

Note that we use the "hat" in $\hat{x}_i^k$ to indicate it is an *estimate* of the measurement x_i^k.

To predict the actual measurement the projected 3D point must be homogenized, which requires dividing by the third homogeneous coordinate. This makes the projection non-linear and precludes using a linear Kalman filter. The filter linearizes the projection equation around the predicted camera system pose to form the Jacobian used in the EKF equations. 3D feature estimates are kept in memory only so long as the feature is tracked by the 2D tracker. This limits the total memory usage of the filter, enabling tracking over large areas. It also means that the filter cannot perform loop completion which is a common limitation of most structure from motion systems when processing long video sequences covering large areas.

One of the main drawbacks of using the EKF is that as the number of tracked salient features increases, the storage space required to store the filter's covariance matrix increases in a quadratic fashion because the matrix stores all of the feature covariances and their cross-covariances. The filter's update cycle complexity is $O(n^3)$ where n is the number of features. This makes real time operation on large sets of features problematic. We avoid this performance bottleneck by taking advantage of the statistical independence of the salient features. So long as the features are stationary with respect to the world coordinate frame, their cross-covariance terms are zero in the filter's covariance matrix. This yields a large, sparse matrix. The structure of this matrix could be exploited to speed up the inversion step, which is part of the Kalman filter.

Another approach is to process the feature measurements, which are taken at the same time, sequentially. This approach to processing in the Kalman filter is described in [2]. The filter update cycle starts by predicting the camera position and covariance at the next time step. Then the filter processes each of the 2D feature measurements in sequential fashion. In each sequential update a subset of the total state comprised of the camera system state and a randomly selected 3D feature estimate is generated and processed to update the filter's state and covariance estimate as well as the position and covariance of the 3D feature. Each 2D feature measurement that is processed reduces the uncertainty of the camera pose a certain amount as well as the uncertainty of the corresponding 3D feature. When processing the features sequentially, features that are processed earlier tend to have a greater influence on the camera pose estimate but only because they cause a correction to the state which later measurements support. So long as features are processed in random order, over time sequential processing can be shown to behave similarly to processing all features at once in a single update cycle [21].

One advantage of sequential processing is that it allows simple outlier detection and rejection. Outliers are detected based on the difference between the estimated 3D point's projection and its corresponding measurement in the current frame. This error is the filter's residual which is an integral component of the Kalman filter. Outliers are not allowed to influence the camera system state and covariance and are removed from the total filter state.

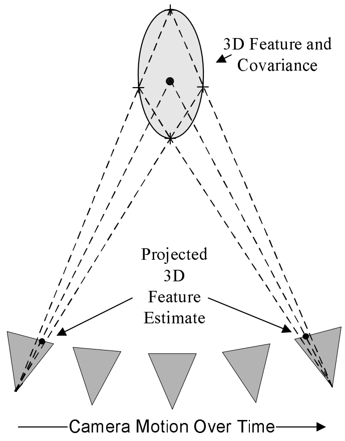


Figure 2: Initial covariance sampling

The initialization of 3D features and their covariances is an important part of the filter design. The filter should strive to initialize 3D estimates for 2D tracks only for inliers. Feature initialization in the leading filter is done by triangulation across a minimum baseline. In addition, the angle between the rays to the feature is measured and a threshold is applied to this angle. In this our filter implementation we chose a threshold of 10^o. This prevents features at infinity from being processed by the filter. (Features at infinity give information about camera rotation but no information about translation and have very large uncertainties, which could cause numerical problems.) Further, the triangulated 3D point is projected into each of the cameras that it has been tracked in 2D so far in the sequence. Only projected features that are within 1 pixel of their corresponding measurements are passed into the filter. This threshold includes both expected measurement error in the KLT tracker, as well as error in the filter's camera pose prediction.

Initial 3D feature covariances are determined by generating a sampled probability distribution in 3D. This is done by intersecting the perturbed rays corresponding to the projected 3D feature estimate in the first frame it is tracked in and the current frame. Each ray is perturbed by the expected amount of measurement noise in eight directions around the projected 3D point in the horizontal, vertical and diagonal directions in the two frames' image spaces. A Gaussian distribution is then fit to this set of samples. A simplified example of this sampling process, sampling only in the horizontal direction, is shown in figure 2.

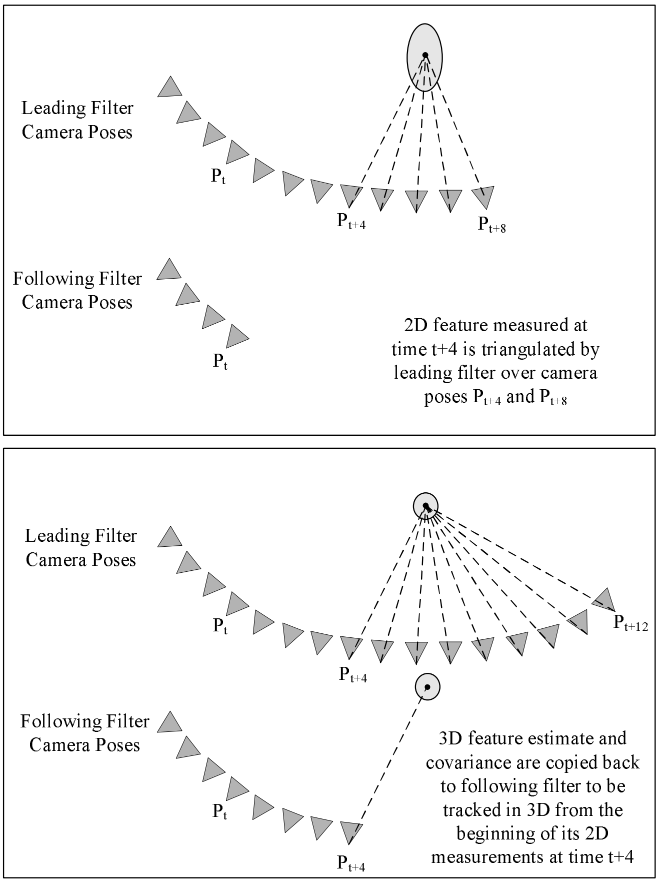
3.2 Two-Stage Extended Kalman Filter Pipeline

In the previous section we described the operation of a single structure from motion process performed by an extended Kalman filter. Our novel approach combines two single filters staggered in time and operating in parallel to improve SfM accuracy. The filter leading in time selects the best set of inliers and initializes their estimated 3D coordinates and uncertainties. Inliers are then passed to the filter following in time, which performs SfM only on the inliers equiped with reliable initial estimates and covariances, improving the SfM accuracy in the following filter.

The leading filter operates on the current frame in the video sequence and selects the best 2D feature tracks, passing initial estimates of the 3D feature locations and feature covariances to the following filter. The following filter operates a fixed number of frames behind the leading filter in the video sequence. Because the following filter receives 3D feature estimates and covariances from the leading filter, it is able to track 2D features from the frame where the 2D track begins and does not have to wait to triangulate the feature or convert feature tracks from a ray/depth/camera center formulation to full 3D formulation, which is done in recent Kalman filter based SLAM implementations [4]. This increases the overall number of good features that are tracked in the following filter each frame.

This two-stage architecture allows a simple and effective form of measurement selection. Features are selected to be passed to the following filter if they are triangulated and then tracked in 3D for a fixed number of frames. Outlier 2D tracks may occasionally be triangulated and added to the leading filter state. However, it is unlikely that these outliers will continue for more than a couple of frames in the leading filter without being rejected as outliers based on their higher reprojection errors due to their inconsistency with the camera motion. By only passing back 3D features that last multiple frames in the leading filter, the following filter processes only features which are consistent with the camera motion. This makes the following filter's camera pose and scene structure estimates more accurate, as demonstrated by reduced reprojection errors. The feature initialization process is shown in figure 3. In that figure dashed lines represent measurements of the 3D feature in a given camera.

Using this architecture the leading and following filters states are not bound together and so the state estimates could drift apart over time. Still both of the cameras' relative motions should be approximately the same over a short time span. The 3D feature locations estimated by the leading filter, which estimates the 3D features in a world coordinate frame, can be passed to the following filter by passing the feature's position relative to the leading camera pose which corresponds to the following filter's current state. In this way the two filters' states are coupled together through the initial 3D feature estimates.


Leading Filter Camera Poses
P_t
P_{t+4}
P_{t+8}
Following Filter Camera Poses
P_t
Leading Filter Camera Poses
P_t
P_{t+4}
P_{t+12}
Following Filter Camera Poses
P_t
P_{t+4}

2D feature measured at time t+4 is triangulated by leading filter over camera poses P_{t+4} and P_{t+8}

3D feature estimate and covariance are copied back to following filter to be tracked in 3D from the beginning of its 2D measurements at time t+4

Figure 3: Initialization of a feature in the leading filter and passing the feature back to the following filter

Our pipelined estimation approach is considerably more efficient that a naive lookahead filter implementation. A naive implementation when estimating the camera pose and scene structure at frame i would process all of the measurements for frames i to $i+w$, where w is the number of frames looked ahead, to find the inlier correspondences to integrate into the final estimate at frame i, repeating this process of looking ahead w frames and then taking a single step at each frame. This would yield an overall computational complexity of $O(wh)$ where w is the number of frames looked ahead and h is the cost of performing one complete SfM estimation on the video sequence. In contrast, our two filter approach is able to determine which correspondences are reliable inliers with a computation cost of only $O(h)$.

4 Results

To demonstrate the improved performance of our two-stage approach we ran the tracking system over ten seconds of video. The video was collected using a camera with known intrinsic calibration and a field of view of approximately 40^ox30^o, frame rate of 30 frames per second and resolution of 1024x768 pixels. The camera was rigidly coupled to an inertial navigation system which was used to initialize the Kalman filter's velocity and rotational velocity estimates. This was necessary because the Kalman filter formulation we use is tuned for a particular scale of motion and so the initial scaled translation and rotation rates must be known. One could just as easily initialize the filter with a fiducial of known size. Recently, Civera [5] has devised a parametrization of structure from motion estimation for the Kalman filter that does not require scale initialization and that could be used in our two-stage architecture to mitigate this limitation.

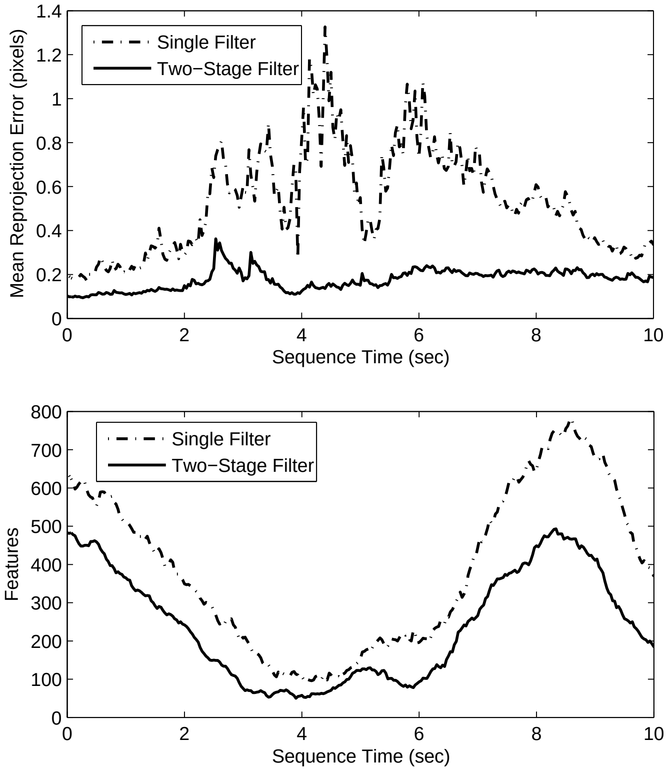


Figure 4: Above: Comparison of mean reprojection error per frame Below: Features included in mean reprojection error per frame

Figure 4 shows the improvement in reprojection errors by selecting measurements and initializing 3D feature estimates using the lookahead filter. The graph shows the mean reprojection errors of all 3D features tracked in each frame, projected into every frame in which they are tracked in 2D. One can clearly see that selecting only those features that are tracked in 3D in the leading filter for 4 or more frames and tracking only those features in the following filter significantly improves the tracking performance of the following filter. No additional non-linear optimization is performed on these results. Figure 4 shows the number of features tracked in 3D using only a single filter which is identical to the leading filter vs. using the two-stage filter architecture. This demonstrates the ability of the two-stage pipelined filter system to select a superior subset of the tracks generated by a single filter system.

5 Conclusion

In this paper we have introduced a measurement selection and initialization approach utilizing a two-stage filter architecture to determine the best set of features and initialize their estimates and uncertainties. These features have lower reprojection errors when processed, allowing for more accurate structure from motion estimation than approaches that attempt to estimate structure from motion in the most recently captured frame with no delay. Pipelined estimation is applicable to many types of robust estimation systems including Kalman and particle filters and is applicable to any system of potentially unreliable sensors, where a reliable set of sensors must be selected and a small delay in estimating the state is acceptable.

Future work on pipelined estimation may involve selecting an optimal set of good features to process (minimal computational cost to process with maximal camera state information) in the following filter which gives a reliable camera pose estimate while minimizing the computational cost of operating multiple filters, allowing for real time filter operation with high accuracy. Additionally, in the current architecture it is possible for the leading filter's scale to drift away from the following filter's over time. Addressing this potential weekness by correcting the leading filter's state using the following filter's more reliable estimates (feeding them forward) would make the system more robust to scale or other drift between the filters.

6 Acknowledgements

We would like to acknowledge the David and Lucille Packard Foundation Fellowship for funding this work.

References

[1] Ali Azarbayejani and Alex P. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):562–575, 1995.

[2] Robert G. Brown and Patrick Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering, 3rd Edition.* John Wiley and Sons, New York, 1997.

[3] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):523–535, 2002.

[4] Javier Civera, Andrew J. Davison, and J. M. M. Montiel. Inverse depth to depth conversion for monocular slam. In *ICRA*.

[5] Javier Civera, Andrew J. Davison, and J. M. M. Montiel. Dimensionless monocular slam. In *Iberian Conference on Pattern Recognition and Image Analysis*, 2007.

[6] A. Davison. Real-time simultaneous localisation and mapping with a single camera. 2003.

[7] Ethan Eade and Tom Drummond. Scalable monocular slam. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 469–476, Washington, DC, USA, 2006. IEEE Computer Society.

[8] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[9] B.D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Int. Joint Conf. on Artificial Intelligence*, pages 674–679, 1981.

[10] P. Meer. *Robust techniques for computer vision. Emerging Topics in Computer Vision, G. Medioni and S. B. Kang (Eds.)*, pages 107–190. Prentice Hall, 2004.

[11] Peter Meer, Doron Mintz, Azriel Rosenfeld, and Dong Yoon Kim. Robust regression methods for computer vision: a review. *Int. J. Comput. Vision*, 6(1):59–70, 1991.

[12] J.Ṽ. Miller and C.Ṽ. Stewart. Muse: robust surface fitting using unbiased scale estimates. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pages 300–306, 1996.

[13] H. E. Rauch. Solutions to the linear smoothing problem. *IEEE Transactions on Automatic Control*, 82(4):371–372, 1963.

[14] H. E. Rauch, F. Tung, and C. T. Striebel. On the maximum likelihood estimates for linear dynamic systems. Technical Report 6-90-63-62, Lockheed Missiles and Space Company, Palo Alto, CA, June 1963.

[15] H.E. Rauch, F. Tung, and C.T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal (American Institute of Aeronautics and Astronautics)*, 3(8):1445–1450, August 1965.

[16] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons, Inc., New York, NY, USA, 1987.

[17] J. Shi and C. Tomasi. Good Features to Track. In *Int. Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[18] Charles V. Stewart. Minpran: A new robust estimator for computer vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(10):925–938, 1995.

[19] Chi-Keung Tang, Gerard G. Medioni, and Mi-Suen Lee. Epipolar geometry estimation by tensor voting in 8d. In *ICCV (1)*, pages 502–509, 1999.

[20] Philip H. S. Torr and Andrew Zisserman. Robust computation and parametrization of multiple view relations. In *ICCV*, pages 727–732, 1998.

[21] Greg Welch and Gary Bishop. Scaat: Incremental tracking with incomplete information. In Turner Whitted, editor, *Computer Graphics*, Annual Conference on Computer Graphics and Interactive Techniques, pages 333–344. ACM Press, Addison-Wesley, Los Angeles, CA, USA (August 3-8), siggraph 97 conference proceedings edition, 1997.