# Image Stabilization by Features Tracking

Alberto Censi, Andrea Fusiello, Vito Roberto

Machine Vision Laboratory, Dept. of Mathematics and Informatics
University of Udine, Italy
{censi,fusiello,roberto}@dimi.uniud.it

## Abstract

*This paper describes a technique for image stabilization in video sequences. The warping that compensates for camera's motion is computed from tracked features in the images. In order to cope with moving objects, a robust technique is used to compute homographies. Moreover, the tracking is made more reliable by using the computed warping to help predicting the features' positions. The effectiveness of the motion compensation is demonstrated by constructing mosaic images from the stabilized sequence and by computing the RMS error. An effort has been made to keep the computational cost low and to reduce the frame rate needed for tracking, with the aim to make a real-time implementation viable.*

## 1. Introduction

Image stabilization consists in compensating for the camera motion by applying a suitable transformation (*warping*) to the image. In the stabilized image, scene points are motionless in spite of camera motion. This makes it easier for an operator to select a point or a region, for example.

Following [2, 9, 17] we track a set of features through the sequence, and use their image motion to estimate the stabilizing warping. Other authors [8] use directly image intensities in a coarse-to-fine approach for single region tracking.

We employ a modified version of the tracker described in [12], using a Kalman filter to predict feature's position. We adopt a fast outlier rejection rule (X84), in order to estimate the homography robustly. In this way, a moving object on a static background can be coped with.

The tracking also takes advantage of the global warping computed at each frame, which is used to predict the position of lost features.

Image stabilization is a technique very close to mosaicing; indeed, the stabilized sequence yields a mosaic by a straightforward merging of its frames. Some authors [5, 11, 13, 19] use feature matching to build mosaics. A similar technique is described in this paper, but in the framework of image stabilization. Others [1, 10, 14] use a top-down approach at different resolutions to estimate the image alignment by direct pixel's brightness comparison.

The rest of the paper is organized as follows. In Section 2 we describe how images are stabilized by computing the appropriate warping function. Then, in Section 3 the tracking method is described. Mosaic construction is briefly addressed in Section 4. Section 5 reports some experimental results and conclusions are drawn in Section 6.

## 2. Motion compensation

If the camera is looking at an approximately planar scene (like an aerial view, for example), corresponding image points are linked by a linear projective transformation [1], called *homography* (see [18] for example). In order to compensate for the relative motion of the camera, we need to compute the homographies that map each frame onto a given *reference image*. In the warped images, static scene points are (ideally) motionless. We assume that each frame in the sequence overlaps with the reference one. There is no point in stabilizing an image which does not overlap with the reference one; in this case the latter should be changed.

Let us suppose that point correspondences through the image sequence are given. The homography matrix $\mathbf{M}$ that links two corresponding points $\mathbf{p}_i$ and $\mathbf{p}_j$ in two generic frames $f_i$ and $f_j$ is defined by the following equation:

$$\mathbf{p}_i = \mathbf{M}\mathbf{p}_j. \tag{1}$$

[1] This is true also if points do not lie on a plane, but the camera is rotating around its optical center. In all other cases, two images are not related by a linear projective transformation, since there is an appreciable parallax.

or

$$\begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & 1 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \\ w_j \end{bmatrix} \tag{2}$$

where points are expressed in homogeneous coordinates, that is, we denote 2-D points in the image plane as $(x, y, w)$ with $(x/w, y/w)$ being the corresponding Cartesian coordinates. Each point correspondence generates two equations, then $n \geq 4$ points generates $2n$ linear equations which are sufficient to solve for $\mathbf{M}$. The over-constrained system is easily solved by computing the *pseudo-inverse* of the system matrix via Singular Value Decomposition [4].

To make the stabilization process less sensitive to possible tracker's failures or to features attached to independently moving objects, we employ a robust rejection rule (X84) [6] to identify outliers, that is, features whose motion is in disagreement with the dominant one (the planar motion of the majority of the features). If $\hat{\mathbf{M}}$ is the least squares homography, the residual of the $i$-th feature is defined as

$$r_i = ||\mathbf{p}_i - \hat{\mathbf{M}}\mathbf{p}_j||. \tag{3}$$

Following the X84 rule we discard those points whose residuals differ more than $5.24$ MAD (Median Absolute Deviations) from the median. The value $5.2$ corresponds to about $3.5$ standard deviations. This rejection rule has a breakdown point of $50\%$, i.e., any majority of the data can overrule any minority.

After rejecting outliers, the final homography is computed using the remaining features.

## 3. Features Tracking

In the previous section we assumed that correspondences through the image sequence had been recovered. *Feature tracking* finds matching by selecting image features and tracks the latter as they move from one frame to another. It can be seen as an instance of the general problem of computing the optical flow at relatively sparse image positions. Methods based on two dimensional features (such as corners) have the advantage that the measured image motion is not affected by the *aperture problem* (see for example [16]).

Following Tomasi and Kanade[15], the features that we track are maximum points of the image autocorrelation function, which roughly corresponds to corners[2].

These features are extracted in the first frame (with subpixel precision) and then tracked in every subsequent frame of the sequence using a linear Kalman filter [3] to estimate and predict their trajectory. We are implicitly assuming that the features' motion is almost linear within the sampling time interval, and the experiments confirm this assumption.

Let's consider the frame sequence $f_0, f_1, f_2, \ldots, f_k, \ldots$ acquired by a camera with frame rate $1/\Delta t$. The state vector of the linear Kalman is defined as follows (see for example [16]):

$$\mathbf{x}_k = [\ x_k \quad y_k \quad u_k \quad v_k\ ]^T$$

where $(x_k, y_k)$, and $(u_k, v_k)$ are respectively position and velocity of each feature point in the frame $k$. The state transition matrix and the measurement matrix are given by

$$\Phi = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

respectively (since they are time independent, we omitted the time subscript).

The noise covariance matrices, $\mathbf{Q}$ and $\mathbf{R}$, model the uncertainty affecting the prediction and the uncertainty affecting measurements, respectively. $\mathbf{Q}$ depends on the local image derivatives: the higher the derivative in one direction, the more reliably the feature's motion along that direction can be predicted. $\mathbf{R}$ depends on the correlation value between current and updated features' window. The higher the correlation, the more confident is the displacement measure. Coefficients in $\mathbf{Q}$ and $\mathbf{R}$, have been hand-crafted with a trial and test process.

To the state vector it is associated the state covariance matrix $\mathbf{P}_k$ (updated dynamically) that encodes the uncertainty of the current state; the region of the phase space centered around the estimated state $\hat{\mathbf{x}}$ which contains the true state with a given probability $c^2$ is given by the ellipsoid:

$$(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{P}_k)^{-1}(\mathbf{x} - \hat{\mathbf{x}})^T \leq c^2.$$

In order to find the position of a given feature in the current frame, we take a small window centered on it and search for the minimum of the SSD (Sum of Square Difference) error in a neighborhood of the predicted position. The predicted state of the Kalman filter gives the predicted position and the state covariance matrix gives the ellipsoidal search region. If this matching fails (i.e., its normalized SSD is above a certain threshold) a search in a fixed neighborhood of the position in the previous frame is performed.

If the matching still cannot be found, the feature goes into a particular state called `ghost` (after [12]), and it will be held as it was virtually still present for a short number of

[2] More precisely, these are points where the gradient is sufficiently high in two orthogonal directions.

subsequent frames, after that either it reappears, or it is finally discarded. The duration of the `ghost` period must be chosen reasonably short (three frames in our case): if a feature disappears for a long time, it is not because of noise or brief occlusion, therefore a more sophisticated management would be needed.

Even if the tracker looses one feature, it keeps searching it in a region around the point where this feature would lie if it moved according to the plane homography computed with the other features. In this way we can ideally keep track of all the features extracted in the first frame, without need to run extraction during tracking which is computationally very expensive. This technique assumes that the first frame overlaps with all the others, which is reasonable in a stabilization scenario. It works well in the case of occlusion or illumination changing phenomena that last for a long time.

## 4. Mosaic construction

The effectiveness of the motion compensation is demonstrated by constructing mosaic images from the stabilized sequence. A *mosaic* is a single image obtained by aligning and merging many other images showing a different portion of the same scene.

For an image sequence with $n$ frames a mosaic image can be constructed by placing the reference frame at the center of the mosaic and then adding every new *stabilized* frame. In the framework of mosaicing, this technique is called “frame to mosaic” approach. The others are “frame to frame” and “mosaic to frame” [7]. In the first case the warping parameters are computed between successive frames of the sequence, and then, given a reference frame, the homographies are composed to obtain the alignment between each frame and the reference frame. This could be dangerous because a registration error introduced early in the sequence influences all the subsequent frames too. The “mosaic to frame” technique is used in dynamic applications, when the images must maintain their own coordinate system, which is the opposite of image stabilization.

In order to merge the current frame into the mosaic, gray levels of overlapping pixels needs to be *blended*. Many blending functions could be employed: use always the last frame, use the first frame, compute the mean, median or other functions of the gray levels. In our case, using the last frame seems the most appropriate technique, but due to little misalignment, uncompensated radial distortion and illumination changes, there would be a discontinuity in the correspondence of frame boundaries in the mosaic. Therefore, we used as a blending function a weighted average, such that the weight of a pixel in the frame to be blended decreases with its distance from the center. Because of the averaging, if there are moving objects in the scene, they appear blurred in the mosaic (but not in the stabilized sequence).

## 5. Results and Discussion

Series of experiments have been conducted to check the effectiveness of the algorithm; we report some of them.

Figure 1 shows the frames 0, 60 and 99 (the last) of an aerial view sequence, with the tracked features superimposed. The feature points (corresponding to corners) are marked with ’+’s and the small circle depicted around each of them indicates the region of the image that contains the true position of the feature with a probability of $99.9\%$. These ellipses are drawn from the covariance matrix which the Kalman filter automatically computes for any feature.

Figures 2(a) and 2(b) show the stabilized image at frame 60 and 99 respectively. The white box in the center of the image is the frame of the reference image (frame 0). Note how image objects remain motionless with respect to the reference frame. In order to better appreciate the stabilization effect, a mosaic composed of the stabilized images is shown in figure 2(c).

To have a numerical assessment of the stabilization, we computed the RMS error between gray levels of each frame and the reference frame, for the original and stabilized sequences (Fig. 5). The error for the stabilized sequence is almost constant, while, as expected, the error for the original sequence grows linearly before reaching saturation.

The sequence shown in Figure 3 is interesting because of the appearance of a distracting object, which could lead to a failure, if the homography computation is not robust. Yet, as shown in Fig. 4, the stabilization is effective. This also shows an example where lost features (the ones occluded by the book) are recovered by guessing their position with the global homography. In the global mosaic (Fig.4(c)) the distracting object is blurred, owing both to motion and to the blending function.

Original and stabilized MPEG sequences are available on the web: http://mvl.dimi.uniud.it/research.html .

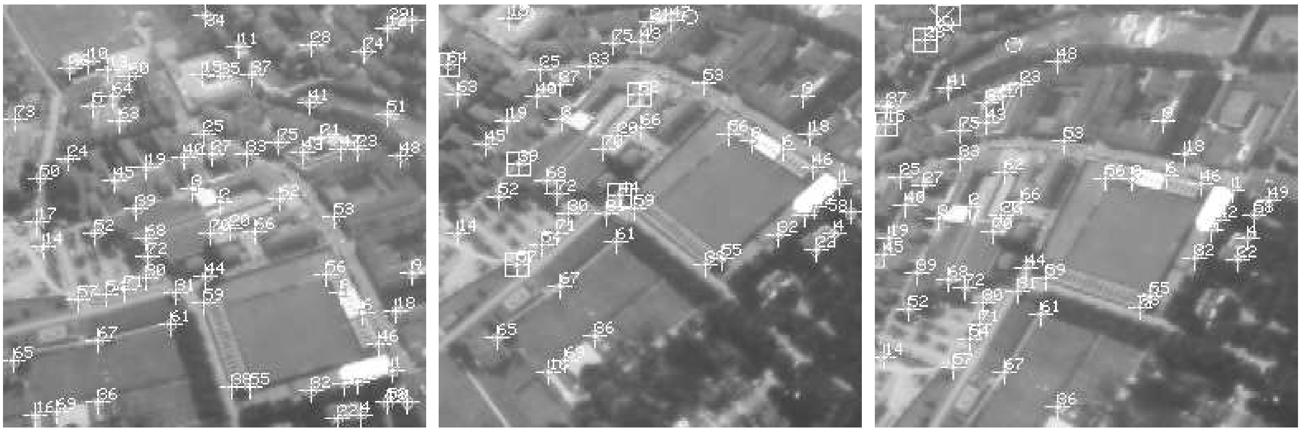
## 6. Conclusions

In this paper we have presented a technique for image stabilization based on feature tracking.

An effort has been made to keep the computational load low. The tracker, based on Kalman filtering, allows for a lower frame rate (i.e.,larger disparity between consecutive frames). The robust technique for computing the homography based on the X84 rejection rule is very efficient, compared to more complicated ones, like LMedS or RANSAC. Moreover, we proposed a new technique for recovering lost features from intermediate frames without running the corner extractor (which is computationally expensive).

Due to a production error, the following three page were omitted from the paper proceedings. The correct electonic version is available from IEEE at http://www.computer.org/proceedings/iciap/0040/0040toc.htm

A causa di un errore della casa editrice, l'articolo apparve negli atti privo delle seguenti tre pagine. La versione elettronica corretta è pubblicata sul sito della IEEE: http://www.computer.org/proceedings/iciap/0040/0040toc.htm

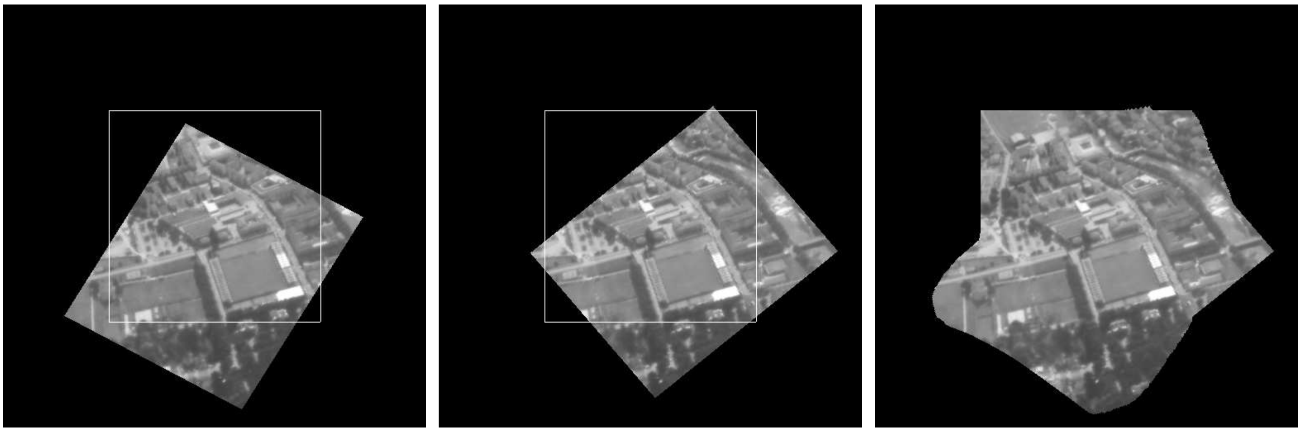


(a) Frame 0

(b) Frame 60

(c) Frame 99

**Figure 1. Some frames from the aerial video sequence. Even if the scene is definitely not planar, the parallax is negligible owing to the great distance from the camera.**

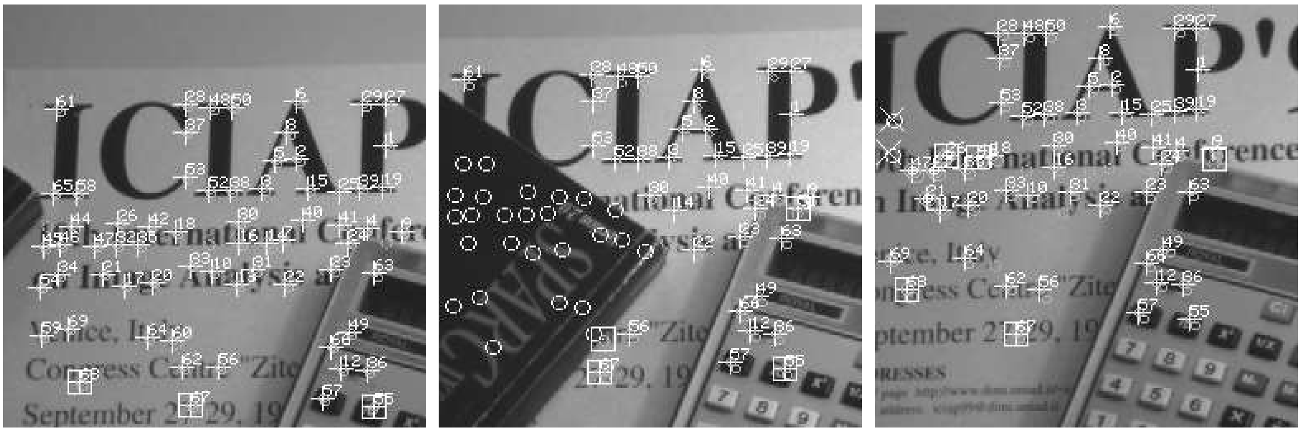


(a) Stabilized frame 60

(b) Stabilized frame 99

(c) Global mosaic

**Figure 2. Stabilized frames of the aerial sequence and global mosaic. The white box is the reference frame, which corresponds to the position of Frame 0.**

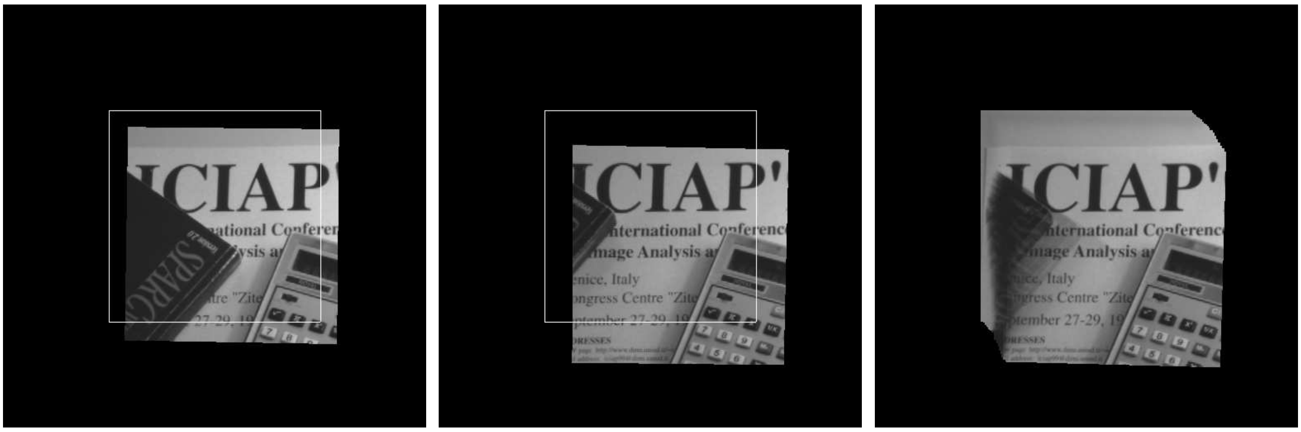


(a) Frame 4

(b) Frame 14

(c) Frame 33

**Figure 3. ICIAP video sequence. In frame 14 the occluding object enters the scene, causing the loss of some features (marked with circles). When the occluding object comes out of the scene, lost features are recovered (marked with $'\times'$).**



(a) Frame 14

(b) Frame 33

(c) Global mosaic

**Figure 4. Stabilized frames of the ICIAP sequence and global mosaic. The white box is the reference frame, which corresponds to the position of Frame 0.**

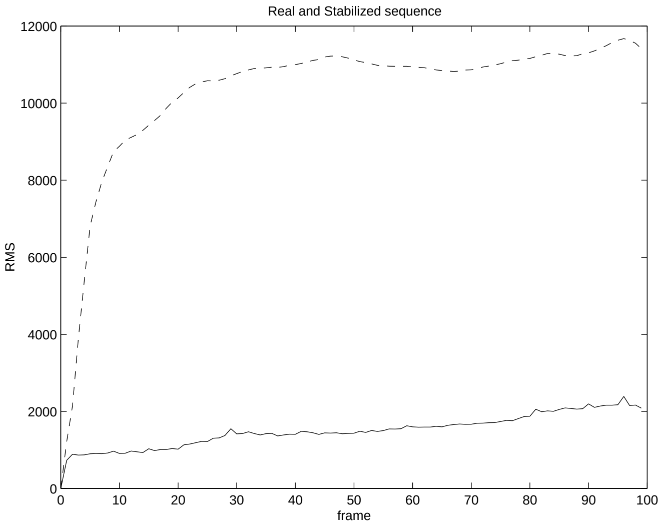


**Figure 5. Root Mean Square (RMS) error for the stabilized aerial sequence (solid line) and the original sequence (dotted line) wrt the reference frame.**

Image stabilization can be seen as a mosaic construction with a frame to mosaic technique. Indeed, we demonstrated the effectiveness of the technique by constructing mosaics from the stabilized images. Quantitative analysis has been made by computing the RMS difference between the stabilized images and the reference frame.

Work is in progress to use warping parameters to set up a fixation control. Following [17] we use the difference between warped and non-warped image centers to drive the position control. Preliminary results, which are not reported here, are encouraging.

## Acknowledgements

This work has been supported by ASI (Agenzia Spaziale Italiana) under contract no. ASI-ARS-96-185. The help of Francesco Isgrò and Jean-Francois Lots in revising the draft is gratefully acknowledged.

## References

[1] T.-J. Cham and R. Cipolla. A statistical framework for long-range feature matching in uncalibrated image mosaicing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 442–447, 1998.

[2] S.-J. Choi, R. R. Schultz, R. L. Stevenson, Y.-F. Huang, and R.-W. Liu. Contrast enhancement of missile video sequences via image stabilization and product correlation. University of Notre Dame, Department of Electrical Engineering, Laboratory for Image and Signal Analysis, http://lisa.ee.nd.edu/rschultz/Papers, 1994.

[3] A. Gelb, editor. *Applied Optimal Estimation*. The M.I.T. Press, 1974.

[4] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The John Hopkins University Press, third edition, 1996.

[5] N. Gracias and J. Santos-Victor. Automatic mosaics creation of the ocean floor. In *Proceedings of the OCEANS Conference*, 1998.

[6] F. Hampel, P. Rousseeuw, E. Ronchetti, and W. Stahel. *Robust Statistics: the Approach Based on Influence Functions*. Wiley Series in probability and mathematical statistics. John Wiley & Sons, 1986.

[7] M. Irani, A. P., J. Bergen, R. Kumar, and S. Hsu. Mosaic representations of video sequences and their applications. *Signal Processing: Image Communication, special issue on Image and Video Semantics: Processing, Analysis, and Application*, 8(4), May 1996.

[8] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image-stabilization. Technical report, The Hebrew Universilty of Jerusalem, Institute of Computer Science, August 1993.

[9] C. Morimoto and R. Chellappa. Fast 3D stabilization and mosaic construction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 660–665, 1997.

[10] H. S. Sawhney and R. Kumar. True multi-image alignment and its application to mosaicing and lens distortion correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 450–456, 1997.

[11] Y. Seo, S. Choi, H. Kim, and K.-S. Hong. Where are the ball and players? Soccer game analysis with color-based tracking and image mosaick. In *Proceedings of the International Conference on Image Analysis and Processing*, pages 196–203, 1997.

[12] L. S. Shapiro, H. Wang, and J. M. Brady. A matching and tracking strategy for independently moving objects. In *Proceedings of the British Machine Vision Conference*, pages 306–315. BMVA Press, 1992.

[13] H. Singh, J. Howland, and D. Yoerger. Quantitative photomosaicking of underwater imagery. In *Proceedings of the OCEANS Conference*, 1998.

[14] R. Szeliski. Image mosaicing for tele-reality applications. Technical report, Cambridge Research Laboratory - Digital Equipment Corporation, 1994.

[15] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, Pittsburg, PA, April 1991.

[16] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall, 1998.

[17] L. Wixson, J. Eledath, M. Hansen, R. Mandelbaum, and D. Mishra. Image alignment for precise camera fixation and aim. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 594–600, 1998.

[18] P. B. Yale. *Geometry and Symmetry*. Dover, 1988.

[19] I. Zoghlami, O. Faugeras, and R. Deriche. Using geometric corners to build a 2D mosaic from a set of images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 420–425, 1997.