Balloon Motion Estimation Using Two Frames*

Qinfen Zheng
SIPI, EEB-400
Univ. of Southern California
Los Angeles, CA 90089

Rama Chellappa
EE Department
Univ. of Maryland
College Park, MD 20742

B.S. Manjunath
ECE Department
Univ. of California
Santa Barbara, CA93106

Abstract

A computational vision approach is presented for the estimation of 2-D translation, rotation, and scale from two partially overlapping images. An illuminant direction estimation method is first used to obtain an initial estimate of camera rotation. A small number of feature points are then located based on a Gabor wavelet model for detecting local curvature discontinuities. An initial estimate of scale and translation is obtained by pairwise matching of the feature points detected from both frames. Finally, hierarchical feature matching is performed to obtain an accurate estimate of translation, rotation and scale. The approach results in a fast method that produces excellent results even when large rotation has occurred between the two frames and the images are devoid of significant features. Experiments with synthetic and real images show that this algorithm yields accurate results when the scale between the image pair differ by up to 10%, *the overlap between the two frames is as small as* 35%, *and the camera rotation between the two frames is significant.*

1 Introduction

Automatic image registration is an important problem in many multi frame based image analysis and applications. Traditional solutions [1, 3, 4, 6, 7, 9] to this problem are unreliable when the rotation of the camera and scale change between the two frames are significant. Registration becomes even more difficult if the images are devoid of significant features and/or the overlap between the two frames is small. In the Mars '94 project we have such a challenging problem. One of the goals of the Mars '94 project is to measure the 3-D wind velocity in Mars. It is proposed to use a downlooking camera attached to a balloon to measure the motion of the balloon and hence determine the wind speed. Figure 1 illustrates the schematic of this project, in which successive image frames are shuttered at times t_1 and t_2. Balloon motion can be determined by measuring the translation, rotation and scale between the image pair. Due to other technical constraints, only two frames will be available for each location, and the rotation and translation between the images could be significant. Also there is a scale change due to the vertical motion of the balloon. A simple and robust registration algorithm is required for on-board motion estimation systems.

In this paper we present a computational vision approach for the estimation of 2-D translation, rotation, and scale from two partially overlapped images. Figure 2 shows the block diagram of our camera motion estimation algorithm. We notice that the illumination on the Mars surface is from the sun and is constant within the time the image pair is taken. By estimating the illuminant direction in each frame, we can estimate the rotation between the two frames and simplify the matching process. Since the common area between the two frames can be much smaller than the image field and in additionally there is scaling between the two frames, methods based on correlation matching become unreliable. In this work we use a feature based matching technique. First we extract a small number of feature points based on a Gabor wavelet model for local curvature analysis. Since no prior knowledge about the translation is available, an initial estimate of scale and translation is obtained by pairwise matching between the neighbors of feature points detected from both the frames. Subsequently, hierarchical correlation matching is performed to obtain an accurate camera motion estimate. Experiments with several real desert balloon images acquired by JPL show that our algorithm works for all the cases tested. The consistency test based on forward and backward motion estimations shows that estimation is quite accurate with the discrepancy in forward and backward estimation of rotation, translation and scale being less than 0.05°, 0.5 pixel, and 3×10^{-3} respectively [10].

The organization of the paper is as follows: Section 2 formulates the balloon motion estimation as an image registration problem; Section 3 discusses the algorithm. Section 4 presents experimental results on Mojave desert images taken from a flying balloon.

2 Balloon Motion Estimation

Let (x_i, y_i, z_i) be the 3-D coordinates and (X_i, Y_i) be the image frame coordinates, both measured with respect to the position of balloon at time t_i, for $i = 1, 2$. Then the central projection equations are

$$\begin{pmatrix} X_i \\ Y_i \end{pmatrix} = \frac{f}{z_i} \begin{pmatrix} x_i \\ y_i \end{pmatrix}, \qquad \text{for } i = 1, 2 \qquad (1)$$

*The support of the Defense Advanced Research Projects Agency (ARPA Order No. 6989) and the U.S. Army Engineer Topographic Laboratories under Contract DACA 76-89-C-0019 is gratefully acknowledged.

where f is the focal length of the camera. The relation between the two image frames can be approximated by

$$\begin{pmatrix} X_2 \\ Y_2 \end{pmatrix} = s \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} X_1 \\ Y_1 \end{pmatrix} + \begin{pmatrix} \Delta X_2 \\ \Delta Y_2 \end{pmatrix} \quad (2)$$

where $s = \frac{\sqrt{\delta X_2{}^2 + \delta Y_2{}^2}}{\sqrt{\delta X_1{}^2 + \delta Y_1{}^2}} = \frac{\frac{f}{z_2}\sqrt{\delta x_2{}^2 + \delta y_2{}^2}}{\frac{f}{z_1}\sqrt{\delta x_1{}^2 + \delta y_1{}^2}} = \frac{z_1}{z_2}$ is the scaling factor, θ is the rotation angle between the two image frames, and $(\Delta X_2, \Delta Y_2)$ is the image translation measured in the image coordinate of frame t_2. With ΔX_2, ΔY_2, θ, and s determined, the balloon motion $\vec{s} = (\Delta x, \Delta y, \Delta z)$ can be easily determined if the height of the balloon is available using[1]

$$\Delta x = \frac{z_2}{f} \cdot \Delta X_2 \quad (3)$$

$$\Delta y = \frac{z_2}{f} \cdot \Delta Y_2 \quad (4)$$

$$\Delta z = z_2 - z_1 = (1 - s) z_2 \quad (5)$$

3 Algorithm

Step 1:
- Estimation of the illuminant azimuth τ_i from frame t_i, $i = 1, 2$;
- Set $\begin{cases} s = 1 \\ \theta = \tau_1 - \tau_2 \\ \Delta X = 0 \\ \Delta Y = 0 \end{cases}$

Step 2:
- Reduce the image size to that of the lowest resolution layer;
- Estimate feature points from each frame.

Step 3:
- Apply an affine transformation with parameters (s, θ, ΔX, ΔY) on the lowest resolution version of frame t_1 and its feature points;
- Do initial matching to obtain estimates (s', θ', $\Delta X'$, $\Delta Y'$);
- Update (s, θ, ΔX, ΔY).

Step 4:
- Reduce the image resolution corresponding to current layer of the matching hierarchy;
- Magnify the coordinates of the feature points corresponding to the resolution of the current layer;
- Apply an affine transform with parameter (s, θ, ΔX, ΔY) on frame t_1;
- Do matching refinement to obtain (s', θ', $\Delta X'$, $\Delta Y'$);
- Update the estimate of (s, θ, ΔX, ΔY).

[1]The height of the balloon can be determined by measuring the time lapse between transmitting and receiving of a radio signal from the balloon.

Step 5:
- If current level is at the highest resolution then stop;
- Increase image resolution and adjust the translation estimation by

$$\begin{pmatrix} 2\Delta X \\ 2\Delta Y \end{pmatrix} \Longrightarrow \begin{pmatrix} \Delta X \\ \Delta Y \end{pmatrix};$$

- Go to Step 4.

Some of the key operations in our algorithm are discussed below.

Initial Estimation of Camera Rotation The initial estimation of camera rotation is computed as the difference of illuminant azimuth angles estimated from both of the frames. To estimate the illuminant direction from image we use the local voting illuminant azimuth estimator reported in [11]. To be more specific, for each pixel we first compute

$$\vec{X} = \begin{pmatrix} \tilde{x}_L \\ \tilde{y}_L \end{pmatrix}$$

by solving

$$\vec{X} = (B^t B)^{-1} B^t d\vec{I}$$

where

$$d\vec{I} = \begin{pmatrix} \delta I_1 \\ \delta I_2 \\ \vdots \\ \delta I_N \end{pmatrix}, \quad B = \begin{pmatrix} \delta x_1 & \delta y_1 \\ \delta x_2 & \delta y_2 \\ \vdots & \vdots \\ \delta x_N & \delta y_N \end{pmatrix},$$

δI is the increment in intensity along the direction $\vec{s} = (\delta x, \delta y)$, and N is the number of measured directions for $\vec{s}$.

The estimate of illuminant azimuth is then computed as

$$\tau = \arctan\left(\frac{\mathbf{E}_{x,y}\left\{ \frac{\tilde{y}_L}{\sqrt{\tilde{x}_L^2 + \tilde{y}_L^2}} \right\}}{\mathbf{E}_{x,y}\left\{ \frac{\tilde{x}_L}{\sqrt{\tilde{x}_L^2 + \tilde{y}_L^2}} \right\}} \right)$$

After the illuminant azimuth τ_i, $i = 1, 2$ are computed, the initial estimation of camera rotation is computed as

$$\theta_0 = \tau_1 - \tau_2 \quad (6)$$

Feature Detection We use a method which is based on a biologically motivated model for identifying local curvature discontinuities, and makes use of Gabor wavelet decomposition of the image and local scale interactions between features [8]. The basic wavelet function used in our decomposition is of the form

$$\Phi_\lambda(x, y, \theta) = e^{-(\lambda^2 x'^2 + y'^2) + i\pi x'} \quad (7)$$

$$x' = x\cos\theta + y\sin\theta \qquad y' = -x\sin\theta + y\cos\theta$$

where θ is the preferred spatial orientation and λ is the aspect ratio of the Gaussian. For convenience, we

will drop the subscripts in further discussion. In all our experiments, λ is set to 1, and θ is discretized into four orientations.

The corresponding wavelet transformation is obtained by convolving the image data $\mathbf{f}$ with a bank of filters whose responses are simple dilations and translations of the basic wavelet in (7), and are denoted by

$$W_j(x,y,\theta) = \mathbf{f} \bigotimes \Phi(\alpha^j x, \alpha^j y, \theta),\ j = \{0,1,2,..\} \quad (8)$$

Here α denotes the scale parameter. Usually the parameter values used are those corresponding to half-octave ($\alpha = \sqrt{2}$) or octave ($\alpha = 2$). Physically, this transformation detects features in the image such as line and step edges. Biologically, this models the processing by simple cells in the visual cortex of mammals. These features by themselves are not good for applications such as obtaining correspondence. The next stage in our feature detection module involves interactions between these simple features (at different spatial frequencies, within each orientation). This step can be identified with the responses of hypercomplex cells in the visual cortex. Hypercomplex cells exhibit end-inhibition. They are sensitive to oriented lines and step edges of short lengths, and their response decreases if the lengths are increased. Using scale interactions to model these cells was first suggested by Hubel and Wiesel [5], who were the first to discover these cells in the visual cortex. Subsequent anatomical studies have also supported this model [2]. We model these interactions as follows:

$$I_{m,n}(x,y) = \max_{\theta} g(||W_m(x,y,\theta) - \gamma W_n(x,y,\theta)||) \quad (9)$$

where g is a non-linear transformation, such as thresholding or a sigmoid non-linearity, γ is a normalizing factor, and $n > m$. In order to identify features at different scales, one has to consider different scale interactions. The final step is to actually localize these features, and this is done by looking at the local maxima of these feature responses.

Scale Estimation Since the Euclidean distance between the feature points only depends on the scale between the two frames, and is invariant to rotation and translation, the scale factor can be estimated prior to the estimation of other parameters. Assume that the matched feature point pairs are $\{(X'_i, Y'_i) \Rightarrow (\hat{X}_i, \hat{Y}_i)\ i = 1, \cdots, N'\}$ with N' be the number of matched feature pairs. Then

$$(\hat{d}_1 \quad \hat{d}_2 \quad \cdots \quad \hat{d}_{N'}) = s' (d_1 \quad d_2 \quad \cdots \quad d_{N'}) \quad (10)$$

where $\hat{d}_i$ is the distance between the feature i and the center of the matched feature points in frame t_1, and d_i is the distance between the feature i and the center of the matched feature points in frame t_2. The scale factor from frame t_1 to frame t_2 is computed as

$$s' = \frac{\sum_{i=1}^{N'} d_i \cdot \hat{d}_i}{\sum_{i=1}^{N'} d_i \cdot d_i}. \quad (11)$$

Rotation and Translation Estimation Assuming that the matched feature point pairs are $\{(X'_i, Y'_i) \Rightarrow (\hat{X}_i, \hat{Y}_i)\ i = 1, \cdots, N'\}$, the relation between the matched feature pairs is

$$\begin{pmatrix} \hat{X}_i \\ \hat{Y}_i \end{pmatrix} = s' \begin{pmatrix} \cos\theta' & \sin\theta' \\ -\sin\theta' & \cos\theta' \end{pmatrix} \begin{pmatrix} X'_i \\ Y'_i \end{pmatrix} + \begin{pmatrix} \Delta X' \\ \Delta Y' \end{pmatrix} \quad (12)$$

Note θ', being the residual of initial rotation estimation, is very small. By approximating $\cos\theta'$ and $\sin\theta'$ up to linear terms, (12) can be rewritten as

$$\begin{pmatrix} \hat{X}_i \\ \hat{Y}_i \end{pmatrix} = \begin{pmatrix} 1 & \theta' \\ -\theta' & 1 \end{pmatrix} \begin{pmatrix} s'X'_i \\ s'Y'_i \end{pmatrix} + \begin{pmatrix} \Delta X' \\ \Delta Y' \end{pmatrix}, \quad (13)$$

or

$$\vec{A} = B\,\vec{C} \quad (14)$$

where

$$\vec{A} = \begin{pmatrix} \hat{X}_1 - s'X'_1 \\ \hat{Y}_1 - s'Y'_1 \\ \vdots \\ \hat{X}_{N'} - s'X'_{N'} \\ \hat{Y}_{N'} - s'Y'_{N'} \end{pmatrix};\quad B = \begin{pmatrix} s'Y'_1 & 1 & 0 \\ -s'X'_1 & 0 & 1 \\ \vdots & \vdots & \vdots \\ s'Y'_{N'} & 1 & 0 \\ -s'X'_{N'} & 0 & 1 \end{pmatrix};$$

$$\vec{C} = \begin{pmatrix} \theta' \\ \Delta X' \\ \Delta Y' \end{pmatrix}.$$

The vector $\vec{C}$ can then be computed as

$$\vec{C} = (B^t B)^{-1} B^t \vec{A}. \quad (15)$$

Estimation Refinement In our method, at each level of the matching hierarchy, frame t_1 is first transformed using the estimated camera motion parameters s', θ', $\Delta X'$, and $\Delta Y'$ and then matched to frame t_2. The correction on initial estimation is computed using (11) and (15). After that, the total transformation is obtained by combining the two transformations together [10]:

$$\begin{pmatrix} \theta + \theta' \\ s\,s' \\ s'(\cos\theta'\,\Delta X + \sin\theta'\,\Delta Y) + \Delta X' \\ s'(-\sin\theta'\,\Delta X + \cos\theta'\,\Delta Y) + \Delta Y' \end{pmatrix} \Rightarrow \begin{pmatrix} \theta \\ s \\ \Delta X \\ \Delta Y \end{pmatrix} \quad (16)$$

4 Experiments

We have tested our algorithm on several balloon images of the Mojave desert, whose environment is similar to Mars. The input image size is 512 x 512. The image size for the lowest resolution layer in our implementation is 128 x 128. For feature detection, m and n of (9) are set to 2 and 5 respectively. In the experiments, we compute the local maximum of $I_{m,n}$ and only salient points which are maximum among the radius of $r = \alpha^n$ are selected as feature points. Due to space limitation, only two examples are presented[2]. Figure 3 shows a typical experiment on registration of desert images obtained from a flying balloon. (a)

[2]More examples and accuracy analysis are given in [10]

and (b) are the input images. (c) and (d) show the feature points detected, in which the white crosses indicate the locations of feature points. In our experiments, the number of feature points detected from each frame is between 15 to 30. This can be controlled by automatically adjustive the threshold of the local maximum. The estimated parameters of motion from Bal109 to Bal112 are $s = 0.995574$, $\theta = 1.60708°$, $\Delta X = -250.769$, and $\Delta Y = 146.602$. (e) shows the mosaicking of (a) and (b) using the estimated motion parameters. (f) shows the difference between the transformed (a) and (b) with the zero of difference shifted to 128.

Figures 4 shows another experiment on image pair with significant camera rotation. (a) and (b) are the input images. (c) and (d) show the feature points detected from each frame. The estimated camera motion parameters are $s = 0.991534$, $\theta = 57.1240°$, $\Delta X = 234.276$, and $\Delta Y = -78.1810$. (e) shows the mosaicking of (a) and (b) using the estimated motion parameters. The continuity of features across the frame borders illustrates the correctness of matching. (f) shows the difference between the transformed (a) and (b) with the zero of difference shifted to 128.

Acknowledgment

The balloon images were available through the courtesy of the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California.

References

[1] P. E. Anuta, "Spatial Registration of Multispectral and Multitemporal Digital Imagery Using FFT Techniques," *IEEE Trans. Geoscience and Electronics*, Vol. GE-8, pp. 353–368, Oct. 1970.

[2] J. Bolz and C. D. Gilbert, "Generation of End-Inhibition in the Visual Cortex via Interlaminar Connections," *Nature*, Vol. 320, pp. 362–365, Mar. 1986.

[3] R. T. Frankot, "'SAR Image Registration by Multi-Resolution Correlation," in *Proc. SPIE*, vol. 220, pp. 195–203, 1983.

[4] T. S. Huang, *Image Sequence Analysis*, Berlin/Heidelberg: Springer-Verlag, 1981.

[5] D. H. Hubel and T. N. Wiesel, "Receptive Fields and Functional Architecture in Two Nonstriate Visual Areas(18 and 19) of the Cat," *J. Neurophysiology*, Vol. 28, pp. 229–289, Mar. 1965.

[6] C. D. Kuglin *et al.*, "Map Matching Techniques for Terminal Guidance Using Fourier Phase Information," in *Proc. SPIE*, vol. 186, 1979.

[7] C. D. Kuglin and D. C. Hines, "The Phase Correlation Image Alignment Method," in *IEEE Int. Conf. on Cybernetics and Society*, pp. 163–165, 1975.

[8] B. S. Manjunath, *"Perceptual Grouping and Segmentation Using Neural Networks"*, Ph.D. dissertation, Department of Electrical Engineering, University of Southern California, Los Angeles, California, Dec. 1991.

[9] Q. Tian and M. N. Huhns, "Algorithms for Subpixel Registration," *Comput. Vision, Graphics, Image Processing*, Vol. 35, pp. 220–233, Aug. 1986.

[10] Q. Zheng and R. Chellappa, "A Computational Vision Approach to Image Registration," Tech. Rep. CAR-TR-583, Center for Automation Research, University of Maryland, College Park, Sept. 1991.

[11] Q. Zheng and R. Chellappa, "Estimation of Illuminant Direction, Albedo and Shape from Shading," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-13, pp. 680–702, July 1991.

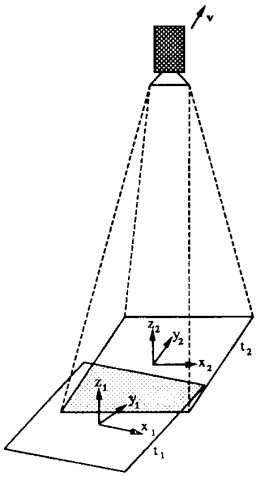

Figure 1: Geometry of Balloon Imagery.

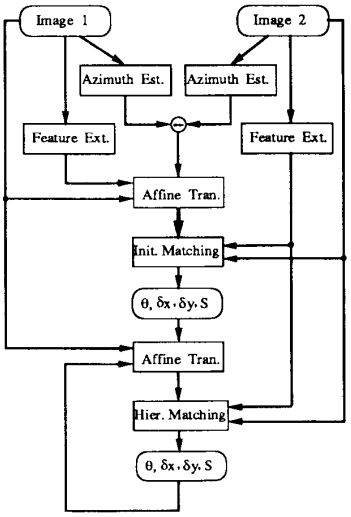

Figure 2: Block diagram of the algorithm

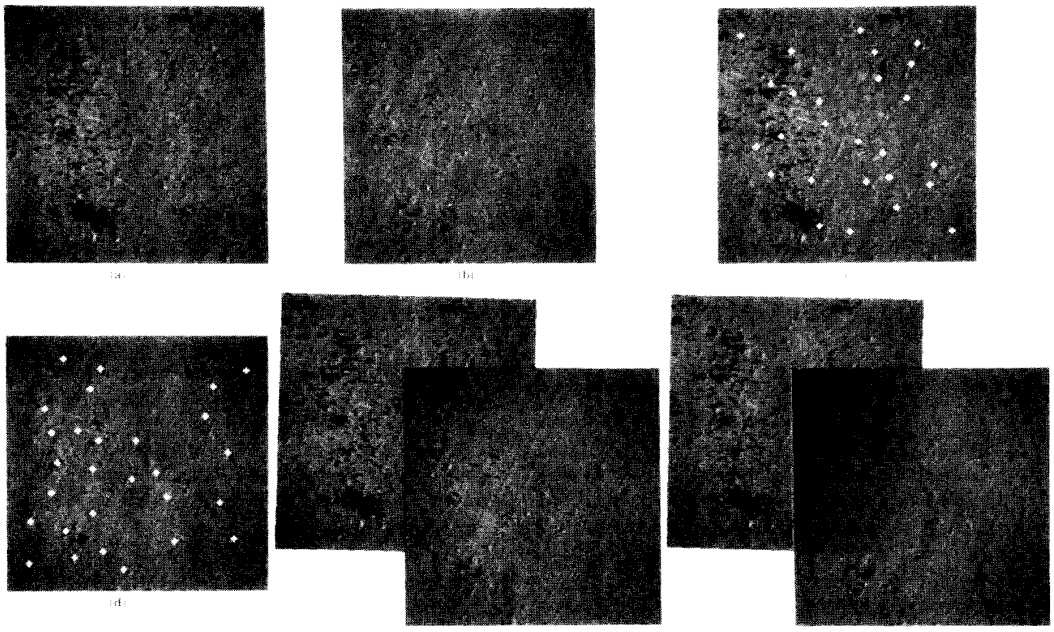

Figure 3: Registration of desert images Bal109 and Bal112.

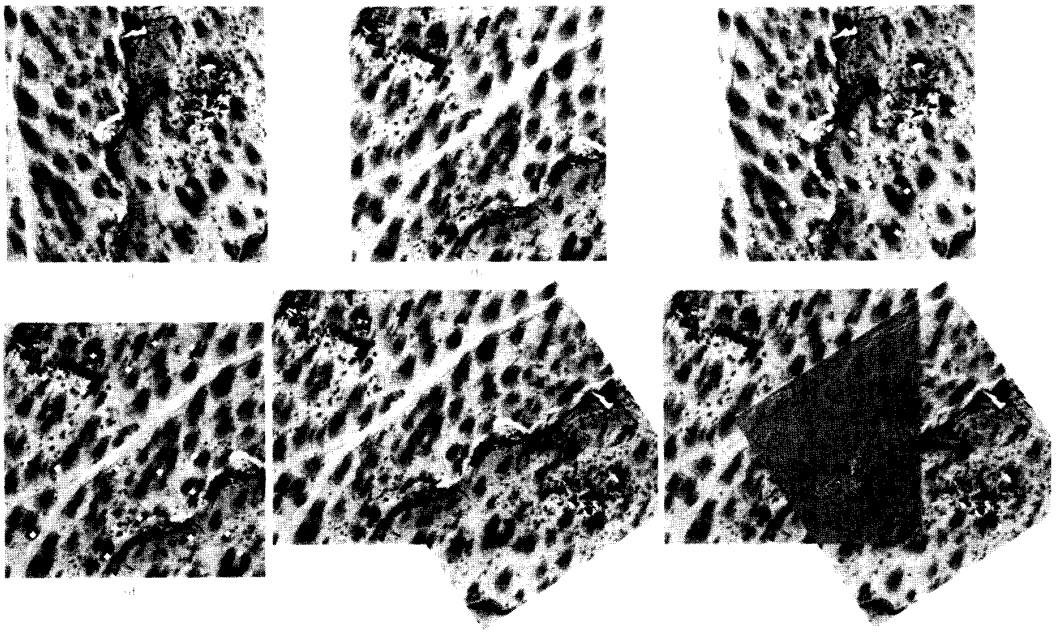

Figure 4: Registration of desert images Bal140 and Bal144.