3.13

Image Sequence Stabilization, Mosaicking, and Superresolution

Rama Chellappa,
S. Srinivasan,
G. Aggarwal, and
A. Veeraraghavan
University of Maryland, College Park

1 Introduction

A sequence of temporal images gathered from a single sensor adds a whole new dimension to two-dimensional (2D) image data. Availability of an image sequence permits the measurement of quantities such as subpixel intensities, camera motion and depth, and detection and tracking of moving objects. In turn, the processing of image sequences necessitates the development of sophisticated techniques to extract this information. With the recent availability of powerful yet inexpensive computers, data storage systems, and image acquisition devices, image sequence analysis has transitioned from an esoteric research domain to a practical area with significant commercial interest.

Motion problems in which the scene motion largely conforms to a smooth, low-order motion model are termed *global motion problems*. Electronic stabilization of video, creating mosaics from image sequences and performing motion superresolution are examples of global motion problems. Applications of these processes are often encountered in surveillance, navigation, tele-operation of vehicles, automatic target recognition (ATR) and forensic science. Reliable motion estimation is critical to these tasks, which is particularly challenging when the sequences display random as well as highly structured systematic errors. The former is primarily a result of sensor noise, atmospheric turbulence and lossy compression, while the latter is caused by occlusion, shadows and independently moving foreground objects. The goal in global motion problems is to maintain the integrity of the solution in the presence of both types of errors.

Temporal variation in the image luminance field is caused by several factors including camera motion, rigid object motion, nonrigid deformation, illumination and reflectance change, and sensor noise. In several situations, it can be assumed that the imaged scene is rigid, and temporal variation in the image sequence is only due to camera and object motion. Classic motion estimation characterizes the local shifts in the image luminance patterns. The global motion that occurs across the entire image frame is typically a result of camera motion and can often be described in terms of

a low-order model whose parameters are the unknowns. Global motion analysis is the estimation of these model parameters.

The computation of global motion has seldom attained the center stage of research due to the (often incorrect) assumption that it is a linear or otherwise well-conditioned problem. In practice, an image sequence displays phenomena that voids the assumption of Gaussian noise in the motion field data. The presence of moving foreground objects or occlusion locally invalidates the global motion model, giving rise to outliers. Robustness to such outliers is required of global motion estimators. Researchers [1–16] have formulated solutions to global motion problems, usually with an application perspective. These can be broadly classified as *feature-based* [6–9], and *flow-based* [1–5] techniques. Feature-based methods extract and match discrete features between frames, and trajectories of these features are fit to a global motion model. In flow-based algorithms, the optical flow of the image sequence is an intermediate quantity that is used in determining the global motion.

In recent years, research on flight navigation of insects, especially bees and flies, has uncovered a number of different optical cues that insects use for successful navigation and stabilization of their flight. In the next section, we will discuss some of the vision based control mechanisms that these insects use for navigational purposes. Section 3 describes the global motion model and its implications. One optical flow-based algorithm is discussed in Section 4. The three primary applications of this procedure are 2D stabilization, mosaicking, and motion superresolution. These are in turn described in Sections 5, 6, and 7, respectively. A related but theoretically distinct problem, three-dimensioanl (3D) stabilization, is briefly discussed in Section 8.

2 Biologic Motivation: Insect Navigation

Insects are able to fly and navigate in this complex visual world. Despite their relatively small nervous system with very few neurons when compared to the human brain, they are still capable of complex tasks such as safe landing, obstacle avoidance, and dead reckoning. Behavioral research with insects suggest that insects primarily use visual information: specifically image motion induced due to ego motion for tackling a number of these navigational tasks. The visual systems of insects differ significantly from the visual system of humans. These differences have profound consequences regarding the nature of visual tasks that insects are adept at. Insects have immobile eyes with fixed focal length. Moreover, they do not possess stereoscopic vision [17]. Insect eyes possess inferior spatial acuity but their eyes sample the world at a significantly higher rate than human eyes do. Moreover their two eyes are also usually placed very close to each other. Because of these differences, insects have evolved to use very different strategies for a range of visual tasks. According to Srinivasan [18], "Vision in insects is a very active process in which perception and action are tightly coupled." In this section we will study this coupling.

A study of how animal visual systems have evolved to perform tasks such as computing optical flow, stabilization, flight control, ego-motion estimation, and so forth, can serve as a motivational tool indicating that such complex tasks can be performed in real-time, with the accuracy desired. In fact, several researchers have used such biologically inspired mechanisms for flight control and obstacle avoidance [19, 20].

2.1 Centering Behavior and Collision Avoidance

Bees that fly through holes (like a window), tend to fly through the center of these holes. Bees, like most other insects, cannot measure distances from surfaces by using stereoscopic vision [17]. Therefore, it is surprising that they are still able to orient themselves at the center of openings. Recent experiments have indicated that bees balance the image motion on the lateral portion of the two eyes as they fly through an opening [21]. When bees were trained to fly in narrow tunnels with certain patterns on the side walls of the tunnels, it was shown [21] that bees tended to fly at the center of this tunnel when the patterns on the side walls were stationary. If one of these patterned side walls was moved in the direction of the bee's flight, thereby reducing the image motion experienced by the bee on that side, the bees moved closer to this side wall. Similarly, when one of the patterned side walls was moved in the direction opposite to the direction of the bee's flight, the bee moved away from the moving wall. This indicates that bees use image motion to indicate the distance from surfaces, lower image motion indicating farther away from the surface and vice versa.

Collision avoidance is another task that is visually driven in most insects. When an insect approaches an obstacle its image expands on it's eyes. Insects are sensitive to this image expansion and turn away from the direction in which the image expansion occurs [18] thereby avoiding collision with obstacles.

2.2 Control of Flight Speed and Stabilization

If some insects are indeed capable of measuring image motion, specifically the angular velocity of the image, then questions as to whether they use this information to control other aspects of their flight arise. In fact experiments reported in [21] and [22] indicate that insects do use estimates of image motion to alter and control their speed of flight. Srinivasan et al. [21] indicate visual control of flight speed in bees is achieved by

monitoring and regulating the apparent image motion, specifically the angular velocity of the image. This modulation of flight speed depending on the image motion has some distinct advantages. This vision-based control of flight speed enables bees to slow down while encountering narrow passages. Bees also owe their ability to flawlessly land on surfaces to this visual feedback–based control system. When a bee is landing on a surface, the bee steadily reduces its forward speed as it approaches the surface. Experiments [21] indicate that forward speed is proportional to altitude, which ensures that the angular velocity of the image of the surface is maintained constant as the bee approaches the surface. This technique ensures that flight speed is close to zero at touchdown and helps in a bee's flawless landing.

Insects also use the image motion to achieve stabilization of their flight. If an insect moving straight is pushed left due to the wind, elementary motion detectors [23] indicate that the image on the front of the retina has moved to the right. The insects generate a counteractive torque to bring them back on course. Studies have indicated that insects use such visual cues in conjunction with other nonvisual sensors like halteres (that act as gyroscopes) in order to stabilize their flight.

2.3 Measuring Distance by Integrating Optical Flow

Some social insects like honeybees for example are able to use visual cues to navigate accurately and repeatedly to food sources that are far away. Moreover, during this flight they are also able to infer the direction and the distance of their food source and reliably communicate it to other bees in their hive. Recent research has indicated that the odometer of the bees is "visually driven" [18]. Experiments have indicated that bees use the extent of image motion in their eyes as a measure of the distance of their flight [21, 24]. This system that estimates distance flown by integrating image motion seems relatively robust to variations in the texture of the environment. Furthermore, it has long been known [25] that honeybees use celestial landmarks to determine the direction of their flight. Interestingly, foraging honeybees are also able to accurately communicate the distance and direction of food sources to other "recruits" through a waggle dance [25].

Thus, insects in general, and honeybees in particular, possess exceptional navigational abilities that are primarily driven by visual feedback from the environment. Moreover insects seem to prefer image motion–based computations to feature based methods. This preference to image motion–based methods can be attributed to the fact that while their eyes possess very little spatial acuity (to infer or extract features), they sample the world at very high rates. Therefore, they are able to estimate image motion and image velocity better. In the succeeding sections we will review some optical flow–based and some feature-based algorithms for stabilization, ego motion estimation, and motion superresolution.

3 Global Motion Models

Prior to examining solutions to the global motion problem, it is worthwhile to verify whether the apparent motion on the image induced by the camera motion can indeed be approximated by a global model. This study takes into consideration the 3D structure of the scene being viewed and its corresponding image. The moving camera has 6 *df*, determining its three translational and rotational velocities. It remains to be seen whether the motion field generated by such a system can be parameterized in terms of a global model largely independent of the scene depth.

The imaging geometry of a perspective camera is shown in Fig. 1. The origin of the 3D coordinate system (X, Y, Z) lies at the optical center C of the camera. The *retinal plane* or *image plane* is normal to the optical axis Z and is offset from C by the focal length f. Images of unoccluded 3D objects in front of the camera are formed on the image plane. The 2D image plane coordinate system (x, y) is centered at the *principal point*, which is the intersection of the optical axis with the image plane. The orientation of (x, y) is flipped with respect to (X, Y) in Fig. 1, due to inversion caused by simple transmissive optics. For this system, the image plane coordinate (x_i, y_i) of the image of the unoccluded 3D point (X_i, Y_i, Z_i) is given by

$$x_i = f\frac{X_i}{Z_i}, \quad y_i = f\frac{Y_i}{Z_i}. \tag{1}$$

The projective relation (1) assumes a rectilinear system, with an isotropic optical element. In practice, the plane containing the sensor elements may be misaligned from the image plane, and the camera lens may suffer from optical distortions including non-isotropy. However, these effects can be compensated by calibrating the camera, and/or remapping the image. In the remainder of this chapter, it is assumed that

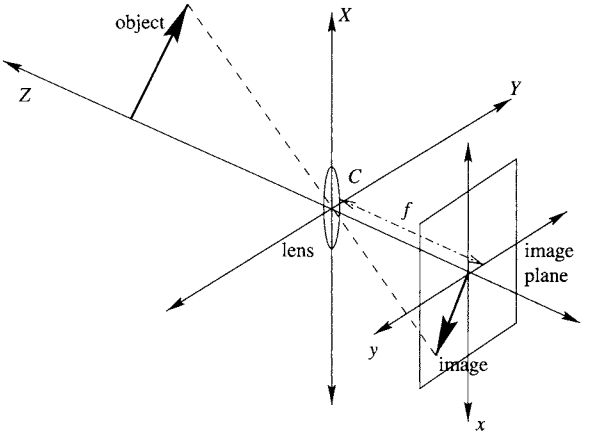


FIGURE 1 Three-dimensional imaging geometry.

the linear dimensions are normalized with respect to the focal length (i.e., $f=1$).

3.1 Feature-based Model

This section presents some intuition behind the applicability of feature-based methods to the problem of global motion estimation. Our basic goal here is to use features to find maps that relate the images taken from different view points. These maps can then probably be used to estimate the involved motion and the model of the scene. Let us take the case of pure rotation. Without loss of generality, the camera center is assumed to be fixed and the image plane is moved to another position. The image of a point in real world is formed by the intersection of the ray joining the camera center to the point with the image planes. Clearly, the images we get on these image planes are quite different but they are related in an interesting way.

Though the lengths, ratios, angles are all different but the *cross-ratio* remains the same [26]. Given four collinear points A, B, C, and D on an image, the cross-ratio is $\frac{AC}{CB} \div \frac{AD}{DB}$ and it remains constant. In other words, $\frac{AC}{CB} \div \frac{AD}{DB} = \frac{A'C'}{C'B'} \div \frac{A'D'}{D'B'}$, where A', B', C', and D' are the corresponding points in the second image (formed after rotating the camera about its axis).

Looking carefully, we can see that this intuition leads to a map relating the two images. Given four corresponding points in general position in the two images, we can map any point from one image to the other. Suppose we know that A maps to A', B to B', C to C', and D to D'. Then the point of intersection of AB and CD (say E) will map to the point of intersection of $A'B'$ and $C'D'$ (say E'). Now any point F on ABE will map to point F' such that the cross-ratio $\frac{AE}{EB} \div \frac{AF}{FB}$ is preserved. This way one can map each point on one image to the other image. Such a map is called *homography*. As mentioned before, such a map is defined by four corresponding points in general position. If x maps to x' by homography H, $x' = Hx$. Note that such a map exist only in case of pure rotation.

However, for planar scenes, homography relating the two views exist regardless of the motion involved. In the case of planar scene, there exist a homography relating the first image to the real-world plane and another one mapping the real-world plane to the second image plane, i.e.,

$$x_1 = H_1 x_p \tag{2}$$

$$x_p = H_2 x_2 \tag{3}$$

$$\Rightarrow x_1 = H_1 H_2 x_2 = H x_2 \tag{4}$$

where H_1 maps x_1, a point on first image plane to x_p, the corresponding point on the real plane while H_2 maps x_p to x_2, the corresponding point on the second image plane. Thus homography $H = H_1 H_2$ maps points from one image plane to the other. Such a homography exists, no matter what the underlying motion between the two camera positions is. This happens because the images formed by camera rotation (or in the case of planar scenes) do not depend on the scene structure. On the other hand, when there are depth variations in the scene, such a homography does not exist between images formed by camera translation.

3.2 Flow-based Model

When a 3D scene is imaged by a moving camera, with translation $t=(t_x, t_y, t_z)$ and rotation $\omega=(\omega_x, \omega_y, \omega_z)$, the optical flow of the scene (Chapter 3.8) is given by

$$\begin{aligned} u(x,y) &= (-t_x + x t_z) g(x,y) + xy\omega_x - (1+x^2)\omega_y + y\omega_z, \\ v(x,y) &= (-t_y + y t_z) g(x,y) + (1+y^2)\omega_x - xy\omega_y - x\omega_z, \end{aligned} \tag{5}$$

for small ω. Here, $g(x, y) = 1/Z(x, y)$ is the inverse scene depth. Clearly, the optical flow field can be arbitrarily complex, and does not necessarily obey a low-order global motion model. However, several approximations to (5) exist that reduce the dimensionality of the flow field. One possible approximation is to assume that translations are small compared with the distance of the objects in the scene from the camera. In this situation, image motion is caused purely by camera rotation, and is given by

$$\begin{aligned} u(x,y) &= xy\omega_x - (1+x^2)\omega_y + y\omega_z, \\ v(x,y) &= (1+y^2)\omega_x - xy\omega_y - x\omega_z. \end{aligned} \tag{6}$$

Equation (6) represents a true global motion model, with 3 *df* (ω_x, ω_y, ω_z). When the field of view (FOV) of the camera is small (i.e., when $|x|, |y| \ll 1$) the second-order terms can be neglected, giving a further simplified three parameter global motion model

$$\begin{aligned} u(x,y) &= -\omega_y + y\omega_z, \\ v(x,y) &= \omega_x - x\omega_z. \end{aligned} \tag{7}$$

Alternatively, the 3D world being imaged can be assumed to be approximately planar. It can be shown that the inverse scene depth for an arbitrarily oriented planar surface is a planar function of the image coordinates (x, y)

$$g(x,y) = ax + by + c. \tag{8}$$

Substituting (8) into (5) gives the eight parameter global motion model

$$u(x,y) = a_0 + a_1x + a_2y + a_6x^2 + a_7xy,$$
$$v(x,y) = a_3 + a_4x + a_5y + a_6xy + a_7y^2, \tag{9}$$

for appropriately computed $\{a_i, i = 0 \ldots 7\}$. Equation (9) is called the *pseudo-perspective* model or transformation.

Equation (5) relating the optical flow with structure and motion assumes that the interframe rotation is small. If this is not the case, the effect of camera motion must be computed using projective geometry [27, 28]. Assume that an arbitrary point in the 3D scene lies at (X_0, Y_0, Z_0) in the reference frame of the first camera, and moves to (X_1, Y_1, Z_1) in the second. The effect of camera motion relates the two coordinate systems according to

$$\begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix} = \begin{pmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{pmatrix} \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \tag{10}$$

where the rotation matrix $[r_{ij}]$ is a function of ω. Combining (1) and (10) permits the expression of the projection of the point in the second image in terms of that in the first as

$$x_1 = \frac{r_{xx}x_0 + r_{xy}y_0 + r_{xz} + t_x/Z_0}{r_{zx}x_0 + r_{zy}y_0 + r_{zz} + t_z/Z_0},$$
$$y_1 = \frac{r_{yx}x_0 + r_{yy}y_0 + r_{yz} + t_y/Z_0}{r_{zx}x_0 + r_{zy}y_0 + r_{zz} + t_z/Z_0}. \tag{11}$$

Assuming either that (a) points are distant compared to the interframe translation (i.e., neglecting the effect of translation) or (b) a planar embedding of the real world (8), the *perspective* transformation is obtained:

$$x_1 = \frac{p_{xx}x_0 + p_{xy}y_0 + p_{xz}}{p_{zx}x_0 + p_{zy}y_0 + p_{zz}},$$
$$y_1 = \frac{p_{yx}x_0 + p_{yy}y_0 + p_{yz}}{p_{zx}x_0 + p_{zy}y_0 + p_{zz}}. \tag{12}$$

The flow field (u, v) is the difference between image plane coordinates $(x_1 - x_0, y_1 - y_0)$ across the entire image. When the FOV is small, it can be assumed that $|p_{zx}x_0|$, $|p_{zy}y_0| \ll |p_{zz}|$. Under this assumption, the flow field, as a function of image coordinate, is given by

$$u(x,y) = \frac{(p_{xx} - p_{zz})x + p_{xy}y + p_{xz}}{p_{zx}x + p_{zy}y + p_{zz}},$$
$$v(x,y) = \frac{p_{yx}x + (p_{yy} - p_{zz})y + p_{yz}}{p_{zx}x + p_{zy}y + p_{zz}}, \tag{13}$$

which is also a perspective transformation, albeit with different parameters. $p_{zz} = 1$, without loss of generality, giving 8 *df* for the perspective model.

Other popular global deformations mapping the projection of a point between two frames are the similarity and affine transformations, which are given by

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = s \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \tag{14}$$

and

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} b_0 \\ b_1 \end{pmatrix}, \tag{15}$$

respectively. Free parameters for the similarity model are the scale factor s, image plane rotation θ, and translation (b_0, b_1). Taking the difference between interframe coordinates of the similarity transform gives the optical flow field model (7) with one constraint on the free parameters. The affine transformation is a superset of the similarity operator, and incorporates shear and skew as well. The optical flow field corresponding to the coordinate affine transform (15) is also a 6-*df* affine model. The perspective operator is a superset of the affine, as can be readily verified by setting $p_{zx} = p_{zy} = 0$ in (12).

The similarity, affine, and perspective transformations are *group operators*, which means that each family of transformations constitutes an equivalence class. The following four properties define group operators:

1. Closure: If $A, B \in \mathcal{G}$ where $\mathcal{G}$ is a group, then the composition $AB \in \mathcal{G}$.
2. Associativity: For all $A, B, C, \in \mathcal{G}$, $(AB)C = A(BC)$.
3. Identity: $\exists I \in \mathcal{G}$ such that $AI = IA = A$.
4. Inverse: For each operator $A \in \mathcal{G}$, there exists an inverse $A^{-1} \in \mathcal{G}$ such that $AA^{-1} = A^{-1}A = I$.

The utility of the closure property is that a sequence of images can be rewarped to an arbitrarily chosen "origin" frame using any single class of operators, and flows computed only between adjacent frames. Since the inverse of each transformation exists, the origin need not necessarily be the first frame of the sequence. Note that the pseudo-perspective transformation (9) is not a group operator. Therefore, to warp an image

under a pseudo-perspective global deformation, it is necessary to register each new image directly to the origin. This can get tricky when the displacement between them is large, worse yet when the overlap between them is small.

In the process of global motion estimation, each data point is the optical flow at a specified pixel, described by the data vector (u, v, x, y). For the affine and pseudo-perspective transformations, it is obvious that the unknowns form a set of linear equations with coefficients that are functions of the data vector components. The same is true for the perspective and similarity operators, although not obvious. For the perspective transform, the denominators of (13) are multiplied out, while for the similarity transform, the substitutions $s_0 = s\cos\theta$ and $s_1 = s\sin\theta$ give rise to linear equations. In particular, the coefficients of the unknowns in the linear equations for the similarity, affine and pseudo-perspective models are functions of the coordinate (x, y) of the data point. Assuming that errors in data are present only in u, v this implies that errors in the linear system for the similarity, affine and pseudo-perspective transforms are present only in the "right-hand side." In contrast, errors exist in all terms for the perspective model. When errors in u, v are Gaussian, the *least squares* (LS) solution of a system of equations of the form (9), (14), or (15) yields the minimum-mean squared error estimate. For the perspective case, the presence of errors in the "left-hand side" calls for a *total least squares* (TLS) [29] approach. In practice, errors in (u, v) are seldom Gaussian, and simple linear techniques are not sufficient.

4 Algorithm

4.1 Feature Based

Section 3.1 uses homography to map one image to the other. As mentioned there, if such a homography exists, four points are sufficient to specify it precisely. The correspondence errors can be handled by using more than four correspondences (if available) in a RANSAC framework [30]. More often than not, neither the scene being viewed is planar nor the motion a pure rotation. In such cases, there is no linear map that relates one image to the other unless one neglects the effect of translation [similar to assumption made in (12)].

Usually, researchers either make assumptions on the basis of domain knowledge or include additional constraints involving more views to take care of the limitations of the geometric approach. Morimoto et al. [31] demonstrate real-time image stabilization that can handle large image displacements based on a 2D multiresolution technique. Avidan and Shashua [32] propose an operation called *threading* that connects two consecutive fundamental matrices using the trifocal tensor as the thread. This makes sure that consecutive camera matrices are consistent with the 3D scene without explicitly recovering it.

All feature-based methods assume that there are features in the images that can reliably be extracted and matched across frames. There are cases when there are hardly any features like in aerial imagery. For such situations, flow-based methods are more suitable.

4.2 Flow Based

The first step in most flow-based stabilization methods is optical flow estimation. The computation of optical flow using image derivatives hinges on the preservation of the image luminance pattern $\psi(x, y, t)$ over time. This translates into the gradient constraint equation (Chapter 3.8 and [33]),

$$\frac{\partial \psi}{\partial t} + u\frac{\partial \psi}{\partial x} + v\frac{\partial \psi}{\partial y} = 0 \quad \forall x, y, t, \tag{16}$$

in the first-order approximation. The flow field (u, v) is a function of location (x, y). For smooth motion fields encountered in typical global motion problems, it is meaningful to model (u, v) as a weighted sum of basis functions

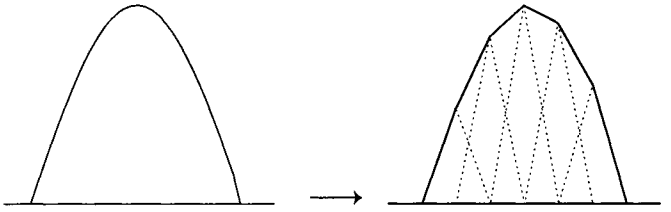
$$u = \sum_{k=0}^{K-1} u_k \phi_k, \quad v = \sum_{k=0}^{K-1} v_k \phi_k. \tag{17}$$

The basis function $\phi_k(x, y)$ is typically a locally supported interpolator generated by shifts of a prototype function $\phi_0(x, y)$ along a square grid of spacing w. An example of linear basis function modeling in 1D is shown in Ex. 1. Additional requirements are imposed on ϕ_0, to ensure computational ease and an intuitive appeal for modeling a flow field. These are:

1. Separability: $\phi_0(x, y) = \phi_0(x)\phi_0(y)$
2. Differentiability: $d\phi_0(x)/dx$ exists $\forall x$.
3. Symmetry about the origin: $\phi_0(x) = \phi_0(-x)$
4. Peak at the origin: $|\phi_0(x)| \leq \phi_0(0) = 1$
5. Compact support: $\phi_0(x) = 0 \quad \forall |x| > w$

The cosine window

$$\phi_0(x) = \frac{1}{2}\left[1 + \cos\left(\frac{\pi x}{w}\right)\right], \quad x \in [-w, w], \tag{18}$$



EXAMPLE 1 *A function (**left**) and its modeled version (**right**). The model used here is the linear interpolator or triangle function. The contribution of each model basis function is denoted by the dotted curves.*

is one such choice of basis that has been shown to accurately model typical optical flow fields associated with global motion problems. A useful range for w is between 8 and 32.

It can be shown that an unbiased estimate for the basis function model parameters $\{u_k, v_k\}$ is obtained by solving the following $2K$ equations [16]

$$\sum_k u_k \int \frac{\partial \phi_k \phi_l \frac{\hat{\partial}\psi}{\partial x}}{\partial x} \psi + \sum_k v_k \int \frac{\partial \phi_k \phi_l \frac{\hat{\partial}\psi}{\partial x}}{\partial y} \psi = \int \phi_l \frac{\hat{\partial}\psi}{\partial t} \frac{\hat{\partial}\psi}{\partial x},$$

$$\sum_k u_k \int \frac{\partial \phi_k \phi_l \frac{\hat{\partial}\psi}{\partial y}}{\partial x} \psi + \sum_k v_k \int \frac{\partial \phi_k \phi_l \frac{\hat{\partial}\psi}{\partial y}}{\partial y} \psi = \int \phi_l \frac{\hat{\partial}\psi}{\partial t} \frac{\hat{\partial}\psi}{\partial y},$$

$$\forall l = 0, 1 \ldots K-1. \tag{19}$$

Each pair of equations of the type (19) characterizes the solution around the image area covered by the basis function ϕ_l. The dominant unknowns, which are the corresponding model weights, are u_l, v_l. The finite support requirement on basis function ϕ_l ensures that only the center weights u_l, v_l and their immediate neighbors in the cardinal and diagonal directions enter each equation. In practice, sampled differentiations and integrations are performed on the sequence. Each equation pair is computed as follows:

1. First, the X, Y and temporal gradients are computed for the observed frame of the sequence. Smoothing is performed before gradient estimation, if the images are dominated by sharp edges.
2. Three templates, each of size $2w \times 2w$, are formed. The first template is the prototype function ϕ_0, with its support coincident with the template. The other two are its X and Y gradients. Knowledge of the analytical expression for ϕ_0 means that its gradients can be determined with no error.
3. Next, a square tile of size $2w \times 2w$ of the original and spatiotemporal gradient images, coincident with the support of ϕ_l is extracted.
4. The 18 left-hand–side terms of each equation and one right-hand–side term are computed by overlaying the templates as necessary and computing the sum of products.
5. Steps 3 and 4 are repeated for all K basis functions.
6. Since the interactions are only between spatially adjacent basis function weights, the resulting matrix is sparse, block tridiagonal, with tridiagonal submatrices, each entry of which is a 2×2 matrix. This permits convenient storage of the left-hand–side matrix.
7. The resulting sparse system is solved rapidly using the preconditioned biconjugate gradients algorithm [34, 35].

The procedure described above produces a set of model parameters $\{u_k, v_k\}$ that largely conforms to the appropriate global motion model, where one exists. In the second phase, these parameters are simultaneously fit to the global motion model while outliers are identified, using the iterated weighted least squares technique outlined below:

1. Initialization:
   (a) All flow field parameters whose support regions show sufficiently large high-frequency energy (quantified in terms of the determinant and condition number of the covariance matrix of the local spatial gradient) are flagged as valid data points.
   (b) A suitable global motion model is specified.
2. Model fitting:
   (a) If there are an insufficient number of valid data points, the algorithm signals an inability to compute the global motion. In this event, a more restrictive motion model must be specified.
   (b) If there are sufficient data points, model parameters are computed to be the LS solution of the linear system relating observed model parameters with the global motion model of choice.
   (c) When a certain number of iterations of this step are complete, the LS solution of valid data points is output as the global motion model solution.
3. Model consistency check:
   (a) The compliance of the global motion model to the overlapped basis flow vectors is computed at all grid points flagged as valid, using a suitable error metric.
   (b) The mean error $\bar{\epsilon}$ is computed. For a suitable multiplier f, all grid points with errors larger than $f\bar{\epsilon}$ are declared invalid.
   (c) Step 2 is repeated.

Typically, three to four iterations are sufficient. Since this system is open-loop, small errors do tend to build up over time. It is also conceivable to use a similar approach to refine the global motion estimate by registering the current image with a suitably transformed origin frame.

It is worthwhile to briefly discuss here a few recently proposed, analytically appealing algorithms for the computation of optical flow. Liu et al. [36] propose a fast multiscale algorithm for dense optical flow estimation. They integrate hierarchic image representation by wavelet decompostion with differential techniques in a novel coarse-and-fine manner. This approach avoids the error propagation from the coarse level that is inherently present in the traditional coarse-to-fine approaches. They show that if a compactly supported wavelet basis with one vanishing moment is carefully selected, hierarchic image, first-order derivative, and corner representation can be obtained from the wavelet decomposation. This way dense optical flow can be estimated using three of the four components of the wavelet decomposition using only two

frames. This automatically takes care of the "flattening-out" problem in traditional pyramid methods, which produce unaccaptable errors when low-texture regions become flat at coarse levels due to blurring.

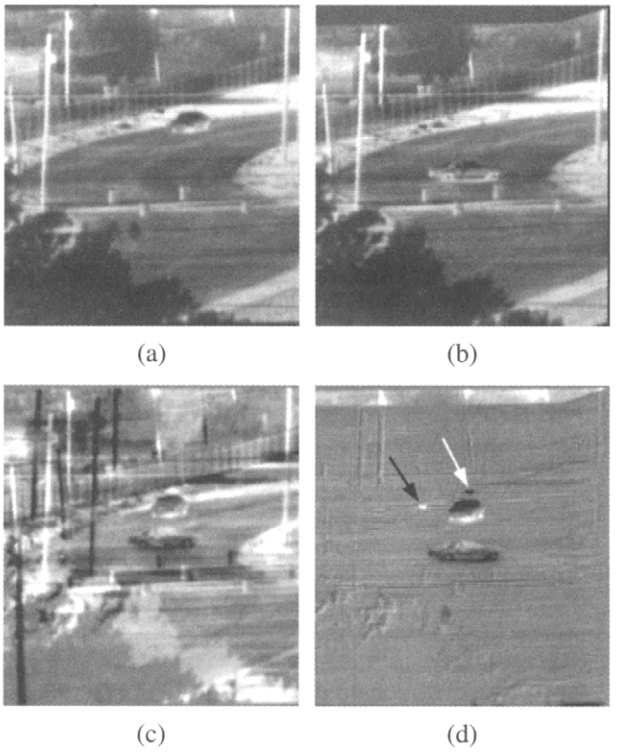
In another work Liu et al. [37] combine the 3D structure tensor with a parametric optical flow model to transform the optical flow estimation problem to a generalized eigenvalue problem. The confidence measure derived from the generalized eigenvalues is used to dynamically adjust the neighborhood to include a wider area of coherent motion. This makes the flow estimation more accurate and robust to aperture problem. An affine model is used as the parametric model for 3D flow instead of the conventional constant flow assumption within the neighborhood.

5 Two-Dimensional Stabilization

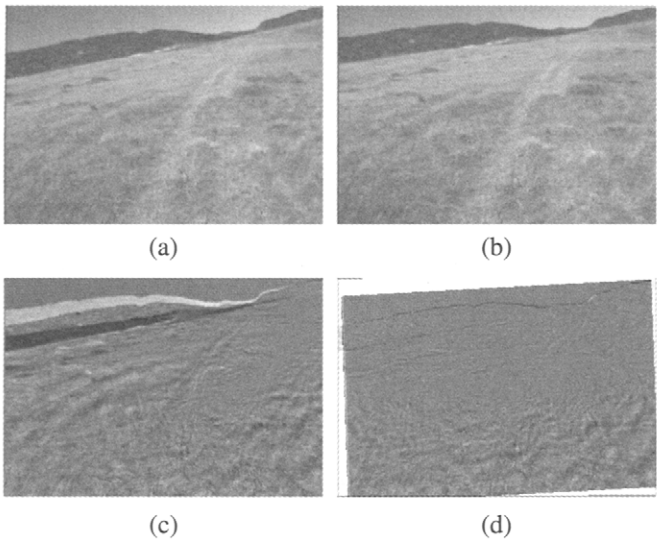
Image stabilization is the process that compensates for the unwanted motion in an image sequence. In typical situations, the term "unwanted" refers to the motion in the sequence resulting from the kinematic motion of the camera with respect to an inertial frame of reference. For example, consider high-magnification handheld binoculars. The jitter introduced by an unsteady hand causes unwanted motion in the scene being viewed. Although this jitter can be eliminated by anchoring the binoculars on a tripod, this is not always feasible. Gyroscopic stabilizers are used by professional videographers, but their bulk and cost are a deterrent to several users. Simpler inertial mechanisms are often found in cheaper "image stabilizing" optical equipment. These work by perturbing the optical path of the device to compensate for unsteady motion. The same effect can be realized in electronic imaging systems by rewarping the generated sequence in the digital domain, with no need for expensive transducers or moving parts.

The unwanted component of motion does not carry any relevant information to the observer, and is often detrimental to subsequent image understanding algorithms. For general 3D motion of a camera imaging a 3D scene, the translational component of the velocity cannot be annulled because of motion parallax. Compensating for 3D rotation of the camera or components thereof is referred to as 3D stabilization, and is discussed in Section 8. More commonly, the optical flow field is assumed to obey a global model, and the rewarping process using the computed global motion model parameters is known as 2D stabilization. Under certain conditions, for example when there is no camera translation, the 2D and 3D stabilization algorithms produce identical results.

The similarity, affine, and perspective models are commonly used in 2D stabilization. Algorithms, such as the one described in Section 4 compute the model unknowns. The interframe transformation parameters are accumulated to estimate the warping with respect to the first or arbitrarily chosen origin frame. Alternatively, the registration parameters of the current frame with respect to the origin frame can be directly estimated. For smooth motion, the former approach allows the use of gradient-based flow techniques for motion computation. However, the latter approach usually has better performance since errors in the interframe transformation tend to accumulate in the former. Two sequences, reflecting disparate operating conditions, are presented here (Ex. 2 and 3) for demonstrating the effect of 2D stabilization. It must be borne in mind that the output of a stabilizer is an image sequence whose full import cannot be conveyed via still images.



EXAMPLE 2 *The first sequence was gathered by a Texas Instruments (TI) infrared camera with a relatively narrow field of view. The scene being imaged is a road segment with a car, a cyclist, two pedestrians, and foliage. The car, cyclist, and pedestrians move across the scene, and the foliage ruffles mildly. The camera is fixated on the cyclist, throwing the entire background into motion. It is difficult for a human observer to locate the cyclist without stabilizing for camera motion. The camera undergoes panning with no rotation about the optical axis and no translation. The first and forty-second frames are shown in (**a**) and (**b**). The difference between these frames with no stabilization is shown in (**c**), with the zero difference offset to 50% gray intensity. Large difference magnitudes can be seen for several foreground and background objects in the scene. On the other hand, the cyclist disappears in the difference image. The same difference, after stabilization, is shown in (**d**). Background areas disappear almost entirely, and all moving foreground objects including the cyclist appear in the stabilized difference. The position of the cyclist in the first and forty-second frames is indicated by the white and black arrows, respectively.*



EXAMPLE 3 *The second image sequence, courtesy of Martin Marietta, portrays a navigation scenario where a forward-looking camera is mounted on a vehicle. The platform translates largely along the optical axis of the camera, and undergoes pitch, roll, and yaw. The camera has a wide field of view, and the scene shows significant depth variation. The lower portion of the image is the foreground, which diverges rapidly as the camera advances. The horizon and distantly situated hills remain relatively static. The third and twentieth frames of this sequence are shown in (**a**) and (**b**). Clearly, forward translation of the camera is not insignificant and full stabilization is not possible. However, the affine model performs a satisfactory job of stabilizing for pitch, roll, and yaw. This is verified by looking at the unstabilized and stabilized frame differences, shown in (**c**) and (**d**), respectively. In (**d**), the absolute difference around the hill areas is visibly very small. The foreground does show change due to forward translation parallax that cannot be compensated for.*

EXAMPLE 4 *This shows a mosaic formed by feature-based method. First of all, features are extracted in each frame followed by correspondence establishment. This is a case of pure rotation. So homgraphy is computed and then all frames are transformed to a common frame of reference.*

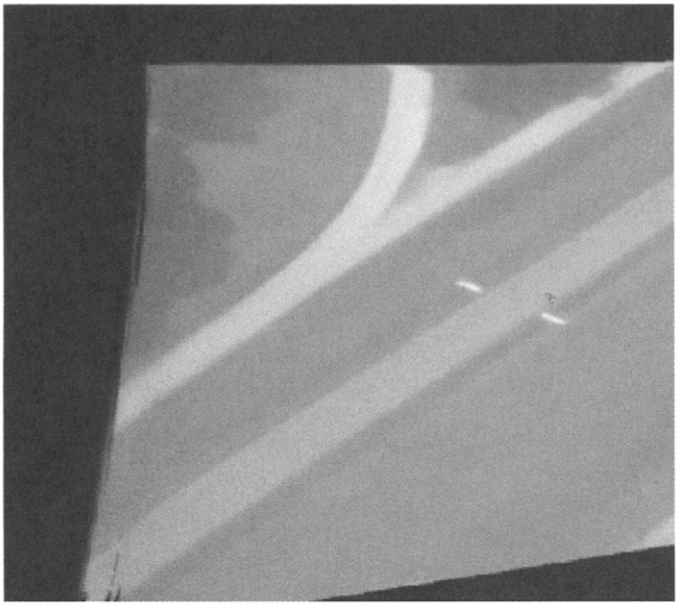
6 Mosaicking

Mosaicking is the process of compositing or piecing together successive frames of an image sequence so as to virtually increase the FOV of the camera [38]. This process is especially important for remote surveillance, teleoperation of unmanned vehicles, rapid browsing in large digital libraries, and video compression. Mosaics are commonly defined only for scenes viewed by a pan/tilt camera. However, recent studies look into qualitative representations, nonplanar embeddings [39], and layered models [40]. The newer techniques permit camera translation and gracefully handle the associated parallax. Mosaics represent the real world in 2D, on a plane or other manifold like the surface of a sphere or "pipe." Mosaics that are not true projections of the 3D world, yet present extended information on a plane are referred to as *qualitative* mosaics.

Several options are available while building a mosaic. A *simple* mosaic is obtained by compositing several views of a static 3D scene from the same view point and different view angles. Two alternatives exist, when the imaged scene has moving objects, or when there is camera translation. The *static* mosaic is generated by aligning successive images with respect to the first frame of a batch, and performing a temporal filtering operation on the stack of aligned images. Typical filters are pixelwise mean or median over the batch of images, which have the effect of blurring out moving foreground objects. Alternatively the mosaic image can be populated with the first available information in the batch.

Unlike the static mosaic, the *dynamic* mosaic is not a batch operation. Successive images of a sequence are registered to either a fixed or a changing origin, referred to as the *backward-* and *forward-stabilized* mosaics, respectively. At any time instant, the mosaic contains all the new information visible in the most recent input frame. The fixed coordinate system generated by a backward-stabilized dynamic mosaic literally provides a snapshot into the transitive behavior of objects in the scene. This finds use in representing video sequences using still frames. The forward-stabilized dynamic mosaic evolves over time, providing a view port with the latest past information supplementing the current image. This procedure is useful for virtual FOV enlargement in the remote operation of unmanned vehicles.

To generate a mosaic, the global motion of the scene is first estimated. This information is then used to rewarp each incoming image to a chosen frame of reference. Rewarped frames are combined in a manner suitable to the end application. The algorithm presented in Section 4 is an efficient means of computing the global motion model parameters. Results using this algorithm are presented in the following examples. Examples 4 and 5 show the mosaics obtained using the feature-based method. In the absence of discernible features, correspondence establishment becomes difficult. The mosaic shown in Ex. 5 contain artifacts due to the errors in correspondence. Examples 6, 7, and 8 illustrate the effectiveness of flow-based methods in such *featureless* scenes.



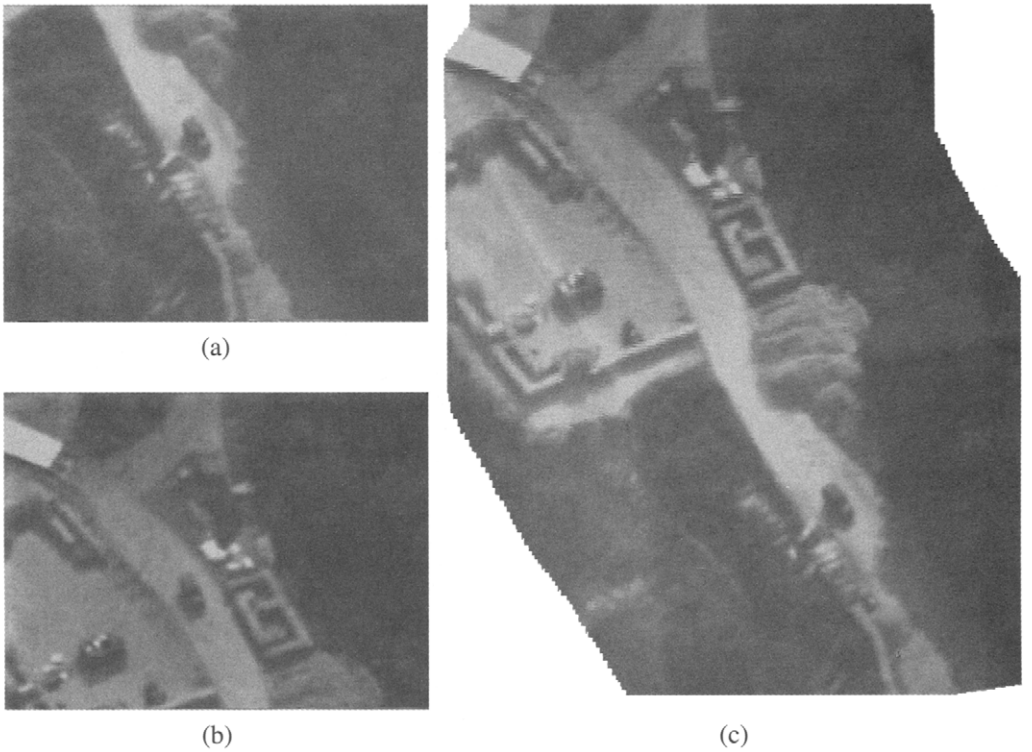
EXAMPLE 5 *This is again a mosaic formed using a feature-based method. In this sequence, there are hardly any features and the motion is arbitrary (although the scene is more or less planar). Few artifacts can be seen along the left edge of the mosaic. This example shows that, as expected, optical flow–based methods perform much better on these types of sequences.*

7 Motion Superresolution

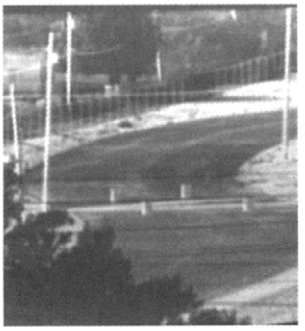
Besides being used to eliminate foreground objects, data redundancy in a video sequence can be exploited for enhancing the resolution of an image mosaic, especially when the overlap between the frames is significant. This process is known as motion superresolution. Each frame of the image sequence is assumed to represent a warped subsampling of the underlying high resolution original. In addition, blur and noise effects can be incorporated into the image degradation model. Let ψ_u represent the underlying image, and $K(x_u, y_u, x, y)$ be a multirate kernel that incorporates the effect of global deformation, subsampling, and blur. The observed low-resolution image ψ is given by

$$\psi(x,y) = \sum_{x_u, y_u} \psi_u(x_u, y_u) K(x_u, y_u, x, y) + \eta(x,y), \qquad (20)$$

where η is a noise process.



EXAMPLE 6 *Images (**a**) and (**b**) show the first and 180th frames of the* Predator F *sequence. The vehicle near the center moves as the camera pans across the scene in the same general direction. Poor contrast is evident in the top right of (**a**), and in most of (**b**). The use of basis functions for computing optical flow pools together information across large areas of the sequence, thereby mitigating the effect of poor contrast. Likewise, the iterative process of obtaining model parameters successfully eliminates outliers caused by the moving vehicle. The mosaic constructed from this sequence is shown in (**c**).*



EXAMPLE 7 *The TI car sequence is reintroduced here to demonstrate the generation of static mosaics. After realignment with the first frame of the sequence, a median filter is applied to the stack of stabilized images, generating the static mosaic shown. Moving objects, viz. the car, cyclist, and pedestrians, are virtually eliminated, giving a pure background image.*

EXAMPLE 8 *This example demonstrates that mosaics can be used to demonstrate dynamic information. This is a 2200-frame mosaic obtained from the predator imagery. Red marks show the path of moving vehicles. The mosaic was created using an iterative variant [41] of the described optical flow–based algorithm. The paths were obtained by appyling Kanade-Lucas Tracker (KLT) tracker [42] on the stabilized sequence (see color insert).*

Example 9 *To illustrate the operation of (20), consider a simple example. Let the observed image be a 4:1 down-sampled representation of the original, with a global translation of (2–3) pixels and no noise. Also assume that the downsampling kernel is a perfect antialiasing alter. The observed image formed by this process is given by*

$$\psi_4(x_u, y_u) = \psi_u(x_u, y_u) * K_0(4x - 2, 4y + 3)$$

$$\mathcal{F}(K_0)(\omega_x, \omega_y) = \begin{cases} 1 & |\omega_x|, |\omega_y| < \frac{\pi}{4} \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

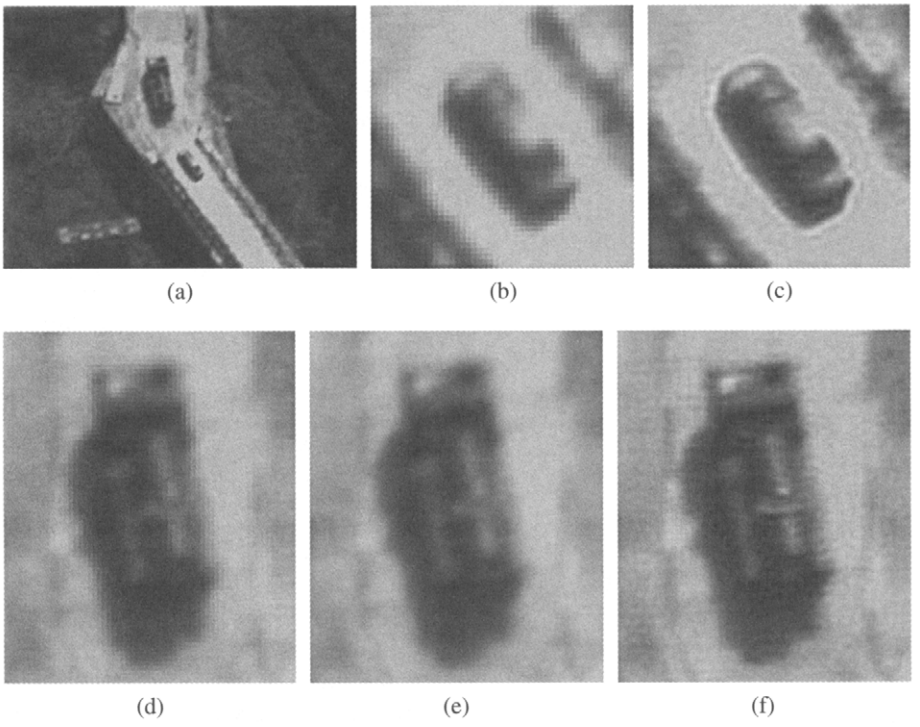
$$\psi(x, y) = \psi_4(4x, 4y),$$

K_0 being the antialiasing alter and $\mathcal{F}(K_0)$ its Fourier transform. The process defined in (21) represents, in some ways, the worst-case scenario. For this case, it can be shown that the original high-pass frequencies can never be estimated, since they are perfectly filtered out in the image degradation process. Thus, multiple high-resolution images produce the same low-resolution images after (21). On the other hand, when the kernel K is a finite support filter, the high-frequency information is attenuated but not eliminated. In theory, it is now possible to restore the original image content, at almost all frequencies, given sufficient low-resolution frames.

Motion superresolution algorithms usually comprise three distinct stages of processing, viz. (*i*) registration, (*ii*) blur estimation, and (*iii*) refinement. Registration is the process of computing and compensating for image motion. More often than not, the blur is assumed to be known, although in theory the motion superresolution problem can be formulated to perform blind deconvolution. The kernel K is specified given the motion and blur. The process of reconstructing the original image from this information and the image sequence data is termed as refinement. Often, these stages are performed iteratively and the high-resolution image estimate evolves over time.

The global motion estimation algorithm outlined in Section 4 can be used to perform rapid superresolution. It can be shown that superresolution can be approximated by first constructing an up-sampled static mosaic, followed by some form of inverse filtering to compensate for blur. This, approximation is valid when the filter K has a high attenuation over its stopband, and thereby minimizes aliasing. Moreover, such a procedure is highly efficient to implement and provides reasonably detailed superresolved frames. Looking into the techniques used in mosaicking, the median filter emerges as an excellent procedure for robustly combining a sequence of images prone to outliers. The superresolution process is defined in terms of the following steps:

1. Compute the global motion for the image sequence.



EXAMPLE 10 *A demonstration of the ability of this relatively simple approach for performing motion superresolution are presented here. The* Predator B *sequence data are gathered from an aerial platform (the predator unmanned air vehicle) and compressed with loss. One frame of this sequence is shown in (**a**). Forty images of this sequence are coregistered using an affine global motion model, up-sampled by a factor of 4, combined and sharpened to generate the superresolved image. **b**, **d**: Car and truck present in the scene, at the original resolution. **e**: The truck image up-sampled by a factor of 4, using a bilinear interpolator. The superresolved images of the car and truck are shown in (**c**) and (**f**), respectively. The significant improvement in visual quality is evident. It must be mentioned here that for noisy input imagery, much of the data redundancy is expended in combating compression noise. More dramatic results can be expected when noise-free input data are available to the algorithm.*

2. For an up-sampling factor M, scale up the relevant global motion parameters.
3. Using a suitable interpolation kernel and scaled motion parameters, generate a stabilized, up-sampled sequence.
4. Build a static mosaic using a robust temporal operator like the median filter.
5. Apply a suitable sharpening operator to the static mosaic.

Example 10 presents a demonstration of motion superresolution.

8 Three-Dimensional Stabilization

Three-dimensional stabilization is the process of compensating an image sequence for the true 3D rotation of the camera. Extracting the rotation parameters for the image sequence under general conditions involves solving the *structure from motion* (SFM) problem, which is the simultaneous recovery of full 3D camera motion and scene structure. Mathematic analysis of SFM shows the nonlinear interdependence of structure and motion given observations on the image plane. Solutions to SFM are based on elimination of the depth field by cross-multiplication [4, 27, 43–46], differentiation of flow fields [47, 48], nonlinear optimization [1, 49], and other approaches. For a comprehensive discussion of SFM algorithms, the reader is encouraged to refer to the literature [16, 27, 28, 50]. Alternatively, camera rotation can be measured using transducers.

Upon computing the three rotation angles, viz. the pitch, roll, and yaw of the camera, the original sequence can be rewarped to compensate for these effects. Alternatively, one can perform *selective stabilization* [51], by compensating the sequence for only one or two of these components. Extending this concept, one can selectively stabilize for certain frequencies of motion so as to eliminate handheld jitter, while preserving deliberate camera pan.

9 Summary

Image stabilization, mosaicking, and motion superresolution are processes operating on a temporal sequence of

images of a largely static scene viewed by a moving camera. The apparent motion observed in the image can be approximated to comply with a global motion model under a variety of circumstances. A simple and efficient algorithm for recovering the global motion parameters is presented here. The 2D stabilization, mosaicking, and superresolution processes are described, and experimental results are demonstrated. The estimation of 2D and 3D motion has been studied for over two decades now, and the following bibliography provides a useful set of starting references for the interested reader.

References

[1] G. Adiv, "Determining 3-d motion and structure from optical flow generated by several moving objects," *IEEE Trans. on Pattern Anal. Mach. Intelli.* **7**, 384–401 (1985).

[2] M. Hansen, P. Anandan, P. J. Burt, K. Dana, and G. van der Wal, "Real-time scene stabilization and mosaic construction," in *DARPA Image Understanding Workshop* (1994), I:457–465.

[3] S. Negahdaripour and B. K. P. Horn, "Direct passive navigation," *IEEE Trans. on Pattern Anal. Mach. Intelli.* **9**, 168–176 (1987).

[4] N. C. Gupta and L. N. Kanal, "3-d motion estimation from motion field," *Arti. Intelli.* **78**, 45–86 (1995).

[5] R. Szeliski and J. Coughlan, "Spline-based image registration," *Int. J. Comput. Vis.* **22**, 199–218 (1997).

[6] Y. S. Yao, *Electronic Stabilization and Feature Tracking in Long Image Sequences*. PhD thesis, University of Maryland, 1996. available as Tech. Rep. CAR-TR-790.

[7] C. Morimoto and R. Chellappa, "Fast 3D stabilization and mosaic construction," in *IEEE Conference on Computer Vision and Pattern Recognition* (San Juan, PR, 1997), 660–665.

[8] H. Y. Shum and R. Szeliski, "Construction and refinement of panoramic mosaics with global and local alignment," in *International Conference on Computer Vision, Mumbai, India* (1998), 953–958.

[9] D. Capel and A. Zisserman, "Automated mosaicing with superresolution zoom," in *IEEE Computer Vision and Pattern Recognition* (Santa Barbara, CA, 1998), 885–891.

[10] M. Irani and S. Peleg, "Improving resolution by image registration," *Graph. Models Image Process.* **53**, 231–239 (1991).

[11] M. S. Alam et al., "High-resolution infrared image reconstruction using multiple randomly shifted low-resolution aliased frames," in *Proc. SPIE* ***3063*** (1997).

[12] R. C. Hardie, K. J. Barnard, and E. E. Armstrong, "Joint map registration and high resolution image estimation using a sequence of undersampled images," *IEEE Trans. Image Process.* **6**, 1621–1633 (1997).

[13] M. Irani, B. Rousso, and S. Peleg, "Recovery of ego-motion using region alignment," *IEEE Trans. Pattern Anal. Mach. Intelli.* **19**, 268–272 (1997).

[14] S. Peleg and J. Herman, "Panoramic mosaics with videobrush," in *DARPA Image Understanding Workshop* (1997), 261–264.

[15] M. Irani and P. Anandan, "Robust multi-sensor image alignment," in *International Conference on Computer Vision, Mumbai, India* (1998), 959–966.

[16] S. Srinivasan, *Image Sequence Analysis: Estimation of Optical Flow and Focus of Expansion, with Applications*. PhD thesis, University of Maryland, 1999.

[17] T. S. Collett and L. I. K Harkness, "Depth vision in animals," *Anal. Vis. Behavi.* MIT Press, Cambridge, MA (1982), 111–176.

[18] M. V. Srinivasan and S. W. Zhang, "Visual motor computations in insects," *Annu. Rev. Neurosci.* **27**, 679–696 (2004).

[19] F. Mura and N. Franceschini, "Visual control of altitude and speed in a flight agent," in *Proceedings of 3rd International Conference on Simulation of Adaptive Behaviour: From Animal to Animals* (1994), 91–99.

[20] T. R. Neumann and H. H. Bulthoff, "Insect inspired visual control of translatory flight," in *Proceedings of the 6th European Conference on Artificial Life ECAL 2001* (2001), 627–636.

[21] M. V. Srinivasan, S. W. Zhang, M. Lehrer, and T. S. Collett, "Honeybee navigation en route to the goal: visual flight control and odometry," *J. Exp. Biol.* **199**, 237–244 (1996).

[22] C. T. David, "Compensation for height in the control of groundspeed by drosophila in a new, "barbers's pole" wind tunnel," *J. Comput. Physiol.* **A147**, 485–493 (1982).

[23] W. Reichardt, "Movement perception in insects," in *Processing of Optical data by Organisms and by Machines*, W. Reichardt, ed. (Academic Press, New York, 1969), 465–493.

[24] H. Esch and J. E. Burns, "Distance estimation by foraging honeybees," *J. Exp. Biol.* **199**, 155–162 (1996).

[25] Von Frisch, *The Dance Language and Orientation of bees*. (Harvard University Press, Cambridge, Massachusetts, 1993).

[26] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision* (Cambridge University Press, Cambridge, UK, 2000).

[27] A. Mitiche, *Computational Analysis of Visual Motion* (Plenum, 1994).

[28] O. D. Faugeras, *Three-Dimensional Computer Vision* (MIT Press, Cambridge, MA, 1993).

[29] S. V. Huffel and J. Vandewalle, in *The Total Least Squares Problem — Computational Aspects and Analysis* (Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1991).

[30] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," in *Comm. ACM* **24**, 381–395 (1981).

[31] C. Morimoto and R. Chellappa, "Fast electronic digital image stabilization for off-road navigation," in *Real-time Imaging* **2**, 285–296 (1996).

[32] S. Avidan and A. Shashua, "Threading fundamental matrices," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**, 73–77 (2001).

[33] C. L. Fennema and W. B. Thompson, "Velocity determination in scenes containing several moving objects," *Comput. Graph. Image Process.* **9**, 301–315 (1979).

[34] O. Axelsson, in *Iterated Solution Methods* (Cambridge University Press, Cambridge, UK, 1994).

[35] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C, 2nd Ed.* (Cambridge University Press, Cambridge, UK, 1992).

[36] H. Liu, R. Chellappa, and A. Rosenfeld, "Fast two-frame multi-scale dense optical flow estimation using discrete wavelet filters," *J. Opti. Soci. Am. A* **20**, 1505–1515 (2003).

[37] H. Liu, R. Chellappa, and A. Rosenfeld, "Accurate dense optical flow estimation using adaptive structure sensors and

a parametric model," *IEEE Trans. Image Process.* **12**, 1170–1180 (2003).

[38] M. Irani, P. Anandan, and S. Hsu, "Mosaic based representations of video sequences and their applications," in *International Conference on Computer Vision, Cambridge, MA* (1995), 605–611.

[39] B. Rousso, S. Peleg, I. Finci, and A. Rav-Acha, "Universal mosaicing using pipe projection," in *International Conference on Computer Vision, Mumbai, India* (1998), 945–952.

[40] J. Y. A. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE Trans. Image Process.* **3**, 625–638 (1994).

[41] F. R. Frigole, in *Robust Stabilization and Mosaicking for Micro Air Vehicle Imagery.* PhD thesis, University of Maryland, 2002.

[42] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition* (1994), 593–600.

[43] R. Y. Tsai and T. S. Huang, "Estimating 3-D motion parameters of a rigid planar patch i," *IEEE Trans. Acoust. Speech Sign. Process.* **29**, 1147–1152 (1981).

[44] X. Zhuang, T. S. Huang, N. Ahuja, and R. M. Haralick, "A simplified linear optical flow-motion algorithm," *Comput. Vis. Graph. Image Process.* **42**, 334–344 (1988).

[45] X. Zhuang, T. S. Huang, N. Ahuja, and R. M. Haralick, "Rigid body motion and the optic flow image," in *First IEEE Conference on AI Applications* (1984), 366–375.

[46] A. M. Waxman, B. Kamgar-Parsi, and M. Subbarao, "Closed-form solutions to image flow equations for 3D structure and motion," *Int. J. Comput. Visi.* **1**, 239–258 (1987).

[47] H. C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proceedings of the Royal Society of London B*, **B-208**, 385–397 (1980).

[48] A. M. Waxman and S. Ullman, "Surface structure and three-dimensional motion from image flow kinematics," *Int. J. Robot. Res.* **4**, 72–94 (1985).

[49] A. R. Bruss and B. K. P. Horn, "Passive navigation," *Comput. Vis. Graph. Image Process.* **21**, 3–20 (1983).

[50] J. Weng, T. S. Hwang, and N. Ahuja, *Motion and Structure from "Image Sequences.* (Springer-Verlag, Berlin, 1991).

[51] Y. S. Yao and R. Chellappa, "Selective stabilization of images acquired by unmanned ground vehicles," *IEEE Trans. Robot. Automat.*, **RA-13**, 693–708 (1997).

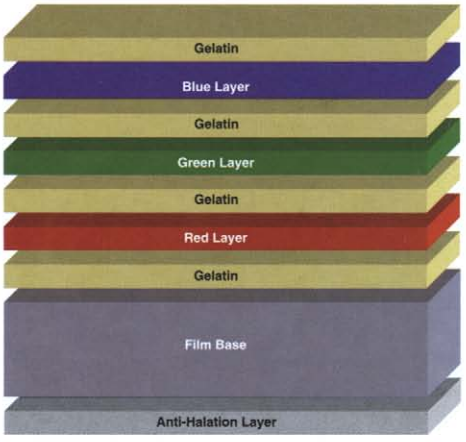


FIGURE 3.11.16 The layered structure of a film.

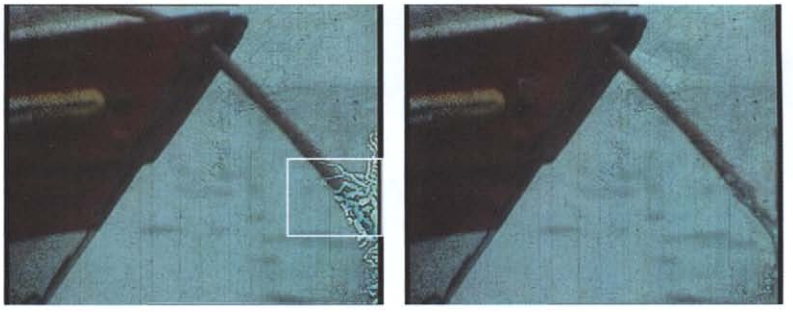


FIGURE 3.11.18 Restoration example (sequence courtesy of RTP — Radiotelevisão Portuguesa). (a) Original frame, with artifact surrounded by a white box; (b) restored frame.

FIGURE 3.13.8 This example demonstrates that mosaics can be used to demonstrate dynamic information. This is a 2,200-frame mosaic obtained from the predator imagery. Red marks show the path of moving vehicles. The mosaic was created using an iterative variant [41] of the described optical flow–based algorithm. The paths were obtained by appyling KLT tracker [42] on the stabilized sequence.