Journal of Mathematical Imaging and Vision 18: 35–54, 2003

Fast Local and Global Projection-Based Methods for Affine Motion Estimation*

DIRK ROBINSON† AND PEYMAN MILANFAR
Department of Electrical Engineering, University of California at Santa Cruz, Santa Cruz, CA 95064, USA
dirkr@ee.ucsc.edu

Abstract. The demand for more effective compression, storage, and transmission of video data is ever increasing. To make the most effective use of bandwidth and memory, motion-compensated methods rely heavily on fast and accurate motion estimation from image sequences to compress not the full complement of frames, but rather a sequence of reference frames, along with "differences" between these frames which results from estimated frame-to-frame motion. Motivated by the need for fast and accurate motion estimation for compression, storage, and transmission of video as well as other applications of motion estimation, we present algorithms for estimating affine motion from video image sequences. Our methods utilize properties of the Radon transform to estimate image motion in a multiscale framework to achieve very accurate results. We develop statistical and computational models that motivate the use of such methods, and demonstrate that it is possible to improve the computational burden of motion estimation by more than an order of magnitude, while maintaining the degree of accuracy afforded by the more direct, and less efficient, 2-D methods.

Keywords: motion estimation, registration, projection, Radon transform, multiscale, affine, performance, complexity

1. Introduction

A fundamental problem in both image sequence processing and computer vision is that of estimating the motion (or dynamics) in an image sequence. For instance, in the field of computer vision, applications of image registration include autonomous navigation, industrial process control, 3-D shape reconstruction, and automatic image sequence analysis. In the field of video coding, the predictive power of accurate motion estimation is used to compress video sequences. In image sequence processing, accurate motion estimates are used to improve overall image resolution. Disparate as they may be, these many applications share one common thread: In all such applications, the computational cost of performing accurate estimation of dynamics is typically very high, and this is often the bottleneck for both performance and real-time implementation. For instance, fast and accurate motion estimation is critical for any real-time motion compensating video encoder. In fact, most real-time video coders require special hardware to achieve the necessary motion estimation efficiency to support real-time encoding.

Dynamic image sequences are modelled as a temporally evolving function $f(x, y, t)$ where x and y represent the spatial coordinates in the image plane and t is the time variable. Written with respect to a reference frame chosen (without loss of generality) at $t = 0$, we then have the model

$$f(x, y, t) = f(x - v_1(x, y)t, y - v_2(x, y)t, 0) \quad (1)$$

where $v_1(x, y)$ and $v_2(x, y)$ denote the components of the velocity (motion) vector field $\vec{v}$. This velocity vector field is sometimes called the optical flow field referring

*This work was supported in part by the National Science Foundation under Grant CCR-9984246.
†To whom correspondence should be addressed.

to the apparent image motion as opposed to the actual motion in the 3-D image scene. The objective of motion estimation is to find the vector field $\vec{v}$, given the image sequence $f(x, y, t)$.

Motion estimation is a widely studied and applied problem. Numerous researchers have developed diverse methods and several survey papers discuss the relative merits of the various leading methods and compare their performances [3, 7, 16, 23].

In this paper, we are concerned with estimating vector fields $\vec{v}$ that are parameterized by an affine model. Namely, the vector fields of interest are characterized by

$$\vec{v} = \vec{v}_0 + M\begin{bmatrix} x \\ y \end{bmatrix}, \tag{2}$$

where

$$\vec{v}_0 = \begin{bmatrix} v_{0x} \\ v_{0y} \end{bmatrix}, \tag{3}$$

is a constant vector representing global translational motion, and

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{4}$$

captures dynamics of rigid body motions as manifested in the image plane.

While there are many methods for estimating affine vector fields, we base our analysis on the popular gradient-based optical flow method. The aim of this paper is to show that these gradient-based methods can be implemented in the Radon transform domain to yield fast, and accurate, estimates of the motion parameters. The Radon transform (projection) of an image is defined as line integrals across the image [10]. It is well-known that pure translational motion in an image results in translation of the projections [10] along the direction of projection. This property has been used successfully in the past to estimate motion using projections [1, 2, 8, 9, 15, 18, 20–22, 24–26]. More recently, we have unified much of the (mostly ad-hoc) work in this area and proposed a model of more general motion vector fields in the Radon transform domain [19]. In particular, it can be shown, as will be elaborated below, that affine motion in the image leads to affine motion in the projections as well.[1] We will use this property to derive efficient and accurate motion estimators using projections.

We further demonstrate that multi-scale implementation of optical flow algorithms using projections yields even more accurate and speedy estimates. The ability to improve computational complexity by almost an order of magnitude makes a compelling case for the routine use of projection-based methods in motion estimation [18, 21, 25].

The paper is organized as follows. Section 2 introduces the direct (2-D) gradient-based method used for estimating affine motion. In Section 3 a novel method for estimating affine motion indirectly using projections is presented. Section 4 describes the application of this indirect (1-D) gradient-based approach in local, global, and multiscale settings. We compare the computational complexity for both the direct and indirect motion estimators in Section 5. Section 6 contains the experimental procedures and results comparing the direct and indirect affine motion estimators. Finally, in Section 7 we conclude with a summary and suggestions for future work.

2. Direct (2-D) Gradient-Based Affine Motion Estimation

A commonly used, and effective method for directly estimating an optical flow field is the gradient based approach. The gradient based methods or differential techniques compute image velocity directly from the image pixel intensities by expanding the right side of (1) in a Taylor series to obtain

$$f(x, y, t) = f(x, y, 0) - v_1 t f_x - v_2 t f_y + \text{higher-order terms} \tag{5}$$

Without loss of generality, we assume that we are examining a pair of images at times $t = 0, 1$ and truncate the Taylor expansion to the first order thereby reducing this expression to the well known gradient constraint equation

$$-f_t = \nabla f \cdot \vec{v}, \tag{6}$$

where $\nabla f = [f_x,\ f_y]^T$ denotes the spatial gradient of f and f_t denotes the difference between two adjacent frames $f(x, y, 1) - f(x, y, 0)$. This constraint can also arise from a more general assumption of intensity conservation where it is assumed that $df/dt = 0$, or the *total* derivative of the image brightness values does not change over some interval of time. Under this intensity conservation assumption, the model of (6) exactly characterizes the optical flow in the image sequence. Under

this assumption f_t becomes the approximation of the partial derivative of the image sequence with respect to time. In general, as the spatio-temporal gradients can be approximated from the given image data, one is able to estimate the desired vector field by assuming that the motion is small, and is constant in some neighborhood. Thus, an overdetermined linear system of equations for the unknowns v_1 and v_2 is arrived at, which can be solved using least-squares or some variant thereof [12].

Returning to the specific case of affine motion and inserting the affine motion model (2) into (6), one obtains a linear equation in the unknown affine motion parameters:

$$-f_t = v_{0x}\ f_x + v_{0y}\ f_y + a\ x\ f_x + b\ y\ f_x + c\ x\ f_y + d\ y\ f_y. \tag{7}$$

To estimate the parameters of motion, we assume that this motion model applies to a spatiotemporal region of the image sequence. Thus, by measuring the gradients in this region (which may in fact be the entire image) we generate a linear system of equations of the form

$$-\mathbf{f}_t = \mathbf{H}\ \mathbf{\Psi} + \mathbf{e} \tag{8}$$

where $\mathbf{f}_t$ denotes the (say raster-scanned) vector of image frame differences measured in the spatio-temporal region of interest, and $\mathbf{e}$ represents noise, or other departures from the assumed model. The matrix $\mathbf{H}$ contains the corresponding spatial gradients at the same points in the region under consideration, and the vector $\mathbf{\Psi}$ is the vector of unknown parameters as follows:

$$\mathbf{H} = [\mathbf{f}_x\ \mathbf{f}_y\ \mathbf{xf}_x\ \mathbf{yf}_x\ \mathbf{xf}_y\ \mathbf{yf}_y] \tag{9}$$

$$\mathbf{\Psi} = [v_{0x}\ v_{0y}\ a\ b\ c\ d]^T \tag{10}$$

In the above, the vectors $\mathbf{xf}_x, \mathbf{xf}_y, \mathbf{yf}_x, \mathbf{yf}_y$ represent the vectors of the raster-scanned values of the partial derivatives $\mathbf{f}_x, \mathbf{f}_y$ weighted by the x or y coordinates. For the purposes of this paper, we assume $\mathbf{e}$ to be a zero-mean white noise. Under this assumption, the best (minimum variance) linear, unbiased estimate of the parameters of interest is given by the least-squares approach [14]:

$$\hat{\mathbf{\Psi}} = -(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{f}_t, \tag{11}$$

$$\mathrm{Cov}(\hat{\mathbf{\Psi}}) = (\mathbf{H}^T\mathbf{H})^{-1}. \tag{12}$$

The assumed noise model is reasonable under the intensity conservation assumption. However, a departure from this assumption can lead to a biased estimator. In practice, even for a reasonably small region (10×10 pixels), this estimator usually provides quite accurate estimates of the affine parameters of the vector field $\vec{v}$. Indeed, the performance of this method and its variations has been studied at some depth in [4–6]. The work of [4] originally outlined the methods for estimating optical flow in a global parametric framework, describing both the models used in this paper for the global translational and global affine model as well as other more complicated models. In [5], the authors propose a region-based optical flow estimation scheme where the blocks are assumed to contain affine motion. Furthermore, the work of [6] explores the use of robust estimators within the context of gradient-based optical flow estimation. While the methods contained in these articles achieve high degrees of accuracy, the computational complexity of the methods is often quite high. The purpose of this paper is to introduce motion estimation using tomographic projections. As we will show, the use of tomographic projections can be incorporated into a variety of motion estimation schemes to achieve substantial speedup with little or no loss in performance. Specifically, we explore the use of projections in gradient-based motion estimation.

3. Using Projections to Estimate Affine Motion

Before we begin the discussion of the use of projections in motion estimation, let us define the Radon transform. The Radon transform [10] of an image $f(x, y)$ is defined as

$$g(p,\theta) = \mathcal{R}_\theta[f(x,y)] = \int\int f(x,y)\delta(p - x\cos\theta - y\sin\theta)\,dx\,dy \tag{13}$$

where δ is the Dirac delta function. A projection of the image can be thought of as the Radon transform evaluated at a particular projection angle θ. As an example, Fig. 1 shows a pair of image projections at 0° and 90°. In this example, the projected image at 0° represents the function created by summing all of the image intensity values in each column of the image. Similarly, the projection at 90° represents the summation of each image row. In general, each point in the projection represents an integration along a line through the original

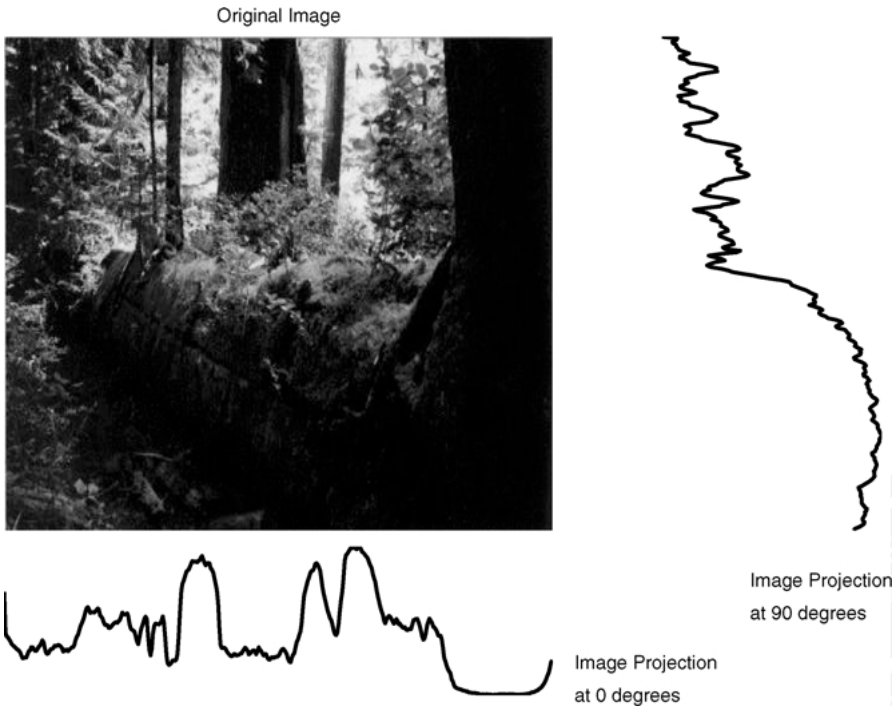

Figure 1. Set of tomographic projections of the forest image.

image. From the definition we see that image projections are symmetric as $g(p, \theta) = g(-p, \theta + \pi)$. We note here that while the above definition represents the model for the Radon transform of a continuous image, we will in practice use a discrete version of the Radon transform.

The use of projections to efficiently estimate motion is not new. Very early works such as [2] use image projections at 0° and 90° to register translated images using a relative phase approach. More recently [15, 25] have incorporated projections into correlation-based block motion estimators to speed up motion compensated video coding. In these works, the projections used to estimate translational motion were confined to 0° and 90°. Similarly, in [8] the authors use correlation between pairs of image projections at 0° and 90° to again register translated images. Furthermore, they find that the use of projection effectively nullifies certain types of pattern noise, yielding improved performance over the direct methods. These works do not, however, address the question of estimating more general image dynamics such as affine motion.

A few researchers have utilized the Radon transform to estimate various forms of affine image motion. The authors of [1, 9, 22] use only a pair of image projections to accelerate motion detection and estimation of a subclass of affine motions, for use in video sequence processing and classification. They constrain the affine motion to that of global magnification and global translation to extract camera movement in video sequences. The work of [26] and [24] describes how the Radon transform could be used to estimate global rotation and translation in image sequences. In particular, [24] uses a set of 360 half image projections or approximately the set of projections at all angles to accurately estimate global rotation and translation for manufacturing process control.

The above methods have not addressed the performance issues concerning the application of projections in estimating both global and local motion, particularly within a multiscale framework. The present work unifies most, if not all the above proposed approaches in a single framework, and establishes a theoretical foundation for their use. In addition, to our knowledge, the present work is the first to justify and use a gradient-based estimation scheme using projections based directly on the analysis of performance vs. computational complexity.

3.1. Motion Under Tomographic Projection

To understand how to estimate motion parameters indirectly using projections, we must first explore the relationship between motion in the original image sequence and the "induced" motion or transformation in the projections. We begin our analysis for the simple case of translational motion which is completely characterized by the shift vector $\vec{v}_0$. The simple relationship known as the shift property of the Radon transform [10], relates motion in images to the motion in projections by

$$\begin{aligned}\mathcal{R}_\theta[f(x - v_{0x}, y - v_{0y})] &= g\big(p - \vec{v}_0^T\vec{w}, \theta\big) \\ &= g(p - u_0(\theta), \theta), \quad (14)\end{aligned}$$

where $\vec{w} = [\cos(\theta), \sin(\theta)]^T$ is a unit direction vector. Intuitively, each projection at angle θ "sees" the component of the vector $\vec{v}_0$ in the direction of the vector $\vec{w}$. Thus, a pure translation or shift given by $\vec{v}_0$ in the image domain results in a corresponding shift in the projection given by $u_0(\theta) = \vec{v}_0^T\vec{w}$.

The question of how general dynamics in image sequences behave under tomographic projection was addressed in [19], where it was shown that under certain smoothness conditions on the image and the vector field $\vec{v}$, for sufficiently small Δt, there exists a unique function $u(p, \theta)$ such that

$$\mathcal{R}_\theta\,[f(x - v_1\Delta t, y - v_2\Delta t)] = g(p - u(p, \theta)\Delta t, \theta) \quad (15)$$

where

$$u(p, \theta)\frac{\partial g(p, \theta)}{\partial p} = \mathcal{R}_\theta[\vec{v}^T\nabla f]. \quad (16)$$

As in [19], we refer to (16) as the Projected Motion Identity (PMI). This relationship suggests that for small transformations (where small depends on the product of the magnitude of the displacement vector field and the time elapsed Δt), the projections of a dynamic image sequence evolve in a qualitatively similar fashion as the original image sequence. That is, the projection function $g(p, \theta)$ evolves as a transformation or warping of the domain coordinates p by the function $u(p, \theta)$. It is important to note here that while the PMI is valid for "small" transformations of the image, it is more universally applicable when applied in a multiscale setting where at coarse scales, large warpings of the image are manifested as small transformations. We elaborate this point further in Section 4.

In the specific case of affine motion, it is shown in [19] that, an affine motion vector field $\vec{v}$ under projection behaves as

$$u(p, \theta) \approx \vec{v}_0^T\vec{w} + (\vec{w}^T M\vec{w})p = u_0(\theta) + \alpha(\theta)p. \quad (17)$$

This suggests that the projected motion $u(p, \theta)$ is also an affine function of the radial parameter p, and is parameterized by $u_0(\theta)$ and $\alpha(\theta)$. We note that the translational component of projected motion depends only on the translational components of the original affine vector field, and the pure linear term also has a corresponding pure linear term in the projection domain. This is part of a more general set of interesting properties of projected motion explored in detail in [19].

For the sake of completeness, it is worth mentioning that the exact form of the affine apparent motion in the projections is known and can be computed using properties of the Radon transform [10]. Namely, the exact form of the projected motion function is

$$u_{exact}(p, \theta) = \vec{v}_0^T\vec{w} + \left(1 - \frac{|\det(J)|}{\|J^T w\|_2}\right)p, \quad (18)$$

where $J = \begin{bmatrix} 1-d & b \\ c & 1-a \end{bmatrix}$ satisfying $(I - M)^{-1} = \frac{1}{|\det(J)|}J$. Comparing (17) and (18), we observe that the only difference appears in the second term. Indeed, as is shown in Appendix B, the term $\alpha(\theta)$ in (17) can be obtained by linearizing the term $(1 - \frac{|\det(J)|}{\|J^T w\|_2})$ in (18) about $M = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$.

In any event, the exact form of the projected motion is highly nonlinear in the parameters of M, and is not easy to use for motion estimation from projections. By contrast, in our approach, we estimate the affine parameters in a *linear* estimation framework. Employing this linear framework, as we will show, has the dual advantage of producing not only very fast, but also quite accurate results.

It is instructive for the affine case to compare the exact formulation to the PMI formulation for a few specific cases:

1. *Pure Scaling.* For the case of pure scaling (e.g. zooming magnification) the affine parameters will have the form $M = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$. Using the exact form of

the projected motion function we obtain

$$u_{exact}(p,\theta) = \left(1 - \frac{|\det(J)|}{\|J^T w\|_2}\right)p = \left(1 - \frac{|1-\lambda|^2}{(|1-\lambda|)\|w\|_2}\right)p = (1-|1-\lambda|)p$$

On the other hand, using the linear form of (17) we obtain

$$u(p,\theta) = (w^T M w)p = \left(w_1^2\lambda + w_2^2\lambda\right)p = \lambda p \tag{19}$$

We observe that for values of λ less than 1, the two equations are equivalent.

2. *Pure Rotation.* For the case of pure rotation by angle ϕ the affine parameters will have the form $M = \left[\begin{smallmatrix} 1-\cos\phi & -\sin\phi \\ \sin\phi & 1-\cos\phi \end{smallmatrix}\right]$. Thus, the exact form of the projected motion function is

$$u_{exact}(p,\theta) = \left(1 - \frac{\det(J)}{\|J^T w\|_2}\right)p = \left(1 - \frac{1}{\|w\|_2}\right)p = 0.$$

This indicates that pure rotation, even in the exact formulation, conveys no information in a single projection. Meanwhile, the PMI approximation yields

$$u(p,\theta) = (w^T M w)p = (1-\cos\phi)p \tag{20}$$

Here we see that the approximation is close to the exact expression for small angles of rotation ϕ. We will again later elaborate on the difficulty of estimating rotation using projections and how this difficulty may be overcome.

3.2. Estimating Projected Motion Parameters

We have just shown that the motion in the projections, or the projected motion, is accurately characterized by the function $u(p,\theta)$ which, in turn, is parameterized by $u_0(\theta)$ and $\alpha(\theta)$. We now present a method for estimating the projected motion parameters $u_0(\theta)$ and $\alpha(\theta)$ from projections at a fixed angle θ over time based on a one-dimensional analog of the optical flow method.

As we did in the derivation of the direct gradient-based estimator, we expand the right side of (15) in a Taylor series truncated to the first order to obtain

$$-g_t = g_p\, u(p,\theta). \tag{21}$$

where g_p denotes the partial derivatives of $g(p,\theta,t)$ with respect to the location variable p and $g_t = g(p,\theta,1) - g(p,\theta,0)$.[2] Interestingly, a corollary of the result (15), proved in [19], is that if the intensity conservation assumption $df/dt = 0$ is invoked in the image domain, the corresponding constraint holds in the projection domain: $dg/dt = 0$. As before, this assumption implies that the model of (21) exactly describes the relationship between image derivatives and image motion. Again, in the context of this assumption g_t refers to the partial derivative of the projected image sequence with respect to time.

Similar to the 2-D case, inserting the affine model (17) into (21) we obtain

$$-g_t = u_0(\theta)g_p + \alpha(\theta)g_p p \tag{22}$$

As in the direct method, we assume the flow applies over some region of the projection, thereby generating an overdetermined system of linear equations,

$$-\mathbf{g}_t(\theta) = \mathbf{H}_1\mathbf{\Psi}_1(\theta) + \boldsymbol{\epsilon}(\theta) \tag{23}$$

where $\mathbf{g}_t(\theta)$ is the a vector containing the temporal differences of $g(p,\theta,t)$ over the area of interest for a particular θ, the matrix $\mathbf{H}_1$ contains the projection spatial derivative information, and the vector $\mathbf{\Psi}_1$ is the vector of unknown parameters as follows:[3]

$$\mathbf{H}_1 = [\mathbf{g}_p\ \mathbf{pg}_p] \tag{24}$$

$$\mathbf{\Psi}_1(\theta) = [u_0(\theta)\alpha(\theta)]^T \tag{25}$$

Again, the notation of $\mathbf{pg}_p$ refers to the vector of the projection partial derivatives weighted by the location indices, and $\mathbf{g}_p$ refers to the vector of unweighted partial derivatives $g_p(p)$. Here, the calculation of the partial derivatives $g_p(p)$ is done in a special fashion that takes into account the geometry of the image region. The discussion of this calculation is presented in Appendix A. It is worth noting here an interesting relationship between the noise $\mathbf{e}$ in the image domain formulation of (8) and the noise $\boldsymbol{\epsilon}(\theta)$ in the corresponding projection domain (23). The noise term $\boldsymbol{\epsilon}(\theta)$ is a projection of the random field $\mathbf{e}$, and as such will still be assumed to be zero-mean. However, assuming the random field comprising the error term $\mathbf{e}$ to be white, with variance σ^2, the noise vector $\boldsymbol{\epsilon}(\theta)$ will have a diagonal covariance matrix $\mathbf{Q}_\theta = \sigma^2\,\mathrm{diag}[S^{-1}(\theta)]$, where the function $S(\theta)$ reflects geometry of the random field region (see Appendix A for further details).

Thus, solving Eq. (23) in a weighted least squares sense we obtain:

$$\hat{\mathbf{\Psi}}_1(\theta) = -\left(\mathbf{H}_1^T \mathbf{Q}_\theta^{-1} \mathbf{H}_1\right)^{-1} \mathbf{H}_1^T \mathbf{Q}_\theta^{-1} \mathbf{g}_t(\theta) \quad (26)$$

$$\mathrm{Cov}(\hat{\mathbf{\Psi}}_1(\theta)) = \left(\mathbf{H}_1^T \mathbf{Q}_\theta^{-1} \mathbf{H}_1\right)^{-1} \quad (27)$$

As before, under the intensity conservation assumption and the zero mean white noise assumption for the original image, (26) is the best linear unbiased estimator for $u_0(\theta)$ and $\alpha(\theta)$.

3.3. *Estimating Affine Motion Parameters from Projected Motion Parameters*

Having described the method for estimating the motion parameters in the Radon transform domain in the previous section, we are now in a position to present the final step in estimating the original parameters of the affine motion model. Namely, the model (17), which relates affine motion in the image domain to the motion in projections can now be invoked. By comparing terms on the left and right-hand sides of (17), we can directly observe that

$$u_0(\theta) = \vec{w}^T \vec{v}_0, \quad (28)$$

$$\alpha(\theta) = \vec{w}^T M \vec{w} \quad (29)$$

This pair of identities allows the estimation of parameters of both the translational part v_0 and the purely linear part M of the vector field $\vec{v}$.

After the projected motion parameters have been estimated at a set of N angles θ_i, $i = 1, \ldots, N$, we can collect all such estimates and write

$$\begin{bmatrix} u_0(\theta_1) \\ \vdots \\ u_0(\theta_N) \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ \vdots & \vdots \\ \cos\theta_N & \sin\theta_N \end{bmatrix} \vec{v}_0 + \varepsilon_0, \quad (30)$$

$$\begin{bmatrix} \alpha(\theta_1) \\ \vdots \\ \alpha(\theta_N) \end{bmatrix} = \begin{bmatrix} \cos^2\theta_1 & \sin^2\theta_1 & 2\cos\theta_1 \sin\theta_1 \\ \vdots & \vdots & \vdots \\ \cos^2\theta_N & \sin^2\theta_N & 2\cos\theta_N \sin\theta_N \end{bmatrix} \times \begin{bmatrix} a \\ d \\ c+b \end{bmatrix} + \varepsilon_m, \quad (31)$$

or equivalently,

$$\mathbf{u}_0 = \mathbf{W}\vec{v}_0 + \varepsilon_0 \quad (32)$$

$$\mathbf{m} = \mathbf{A}\vec{\mu} + \varepsilon_m \quad (33)$$

Because the noise terms ε_0 and ε_α are in general correlated, we combine these estimates into one system of the form

$$\begin{bmatrix} \mathbf{u}_0 \\ \mathbf{m} \end{bmatrix} = \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \vec{v}_0 \\ \vec{\mu} \end{bmatrix} + \varepsilon \quad \text{or} \quad \mathbf{y} = \mathbf{D}\mathbf{\Phi} + \varepsilon \quad (34)$$

where the error vector ε is assumed to be zero-mean with a banded covariance matrix $\mathbf{R}$. The covariance matrix $\mathbf{R}$ is constructed from the collection of covariance matrices of (27).

We estimate $\mathbf{\Phi}$ via weighted least squares:

$$\hat{\mathbf{\Phi}} = (\mathbf{D}^T \mathbf{R}^{-1} \mathbf{D})^{-1} \mathbf{D}^T \mathbf{R}^{-1} \mathbf{y} \quad (35)$$

with corresponding covariance given by

$$\mathrm{Cov}(\mathbf{\Phi}) = (\mathbf{D}^T \mathbf{R}^{-1} \mathbf{D})^{-1}. \quad (36)$$

Ultimately, we will compare the performance of this estimator with that of the original 2-D method presented in Eqs. (11) and (12) of Section 2.

It is important to recall that a drawback of using a projection-based estimator is the inability to directly estimate all of the parameters of M uniquely. Namely, we cannot estimate the component $c - b$ under projection. While the $c + b$ term represents a measure of the shearing of the image sequence, the "missing" term $c - b$ corresponds to the curl of the motion vector field. As we indicated earlier, this suggests that pure rotation will not be distinguishable in a single projection even in the case of the exact projected affine model of (18). At first glance, it would appear that estimating rotational motion is then not at all possible from projections—this is not the case. Indeed, if the complete set of projections of the images were computed, then the angle of rotation could be easily determined by computing pairwise correlation coefficients between a projection (at, say, $\theta = \theta_0$) and the many other available projections. The angle of rotation is then determined by the difference in the projection angles of the pair of projections with highest spatial correlation coefficient. In our method, in order to keep the computational complexity to a minimum, we deal with only a small number of projections

(3 or 4) sampled sparsely in the range $[0, \pi]$, so that the correlation approach is impractical.

Fortunately, our method can still be modified and employed to estimate purely rotational motion. Though we do not pursue this specific problem in this paper, we shall indicate how this can be done by first recalling an important property of projected motion. Let us recall that it was proved in [19], and mentioned earlier in this paper, that projected motion satisfies the a linearity property so that translational motion maps to a single component (u_0) in the projections and the linear part M maps to another separate component ($\alpha(\theta)$) in the projections. This linearity idea can be further exploited to show that the complementary rotational and irrotational components of motion are also separated in the projections. The implication here is that if we simply ignore the fact there is a rotational component in the vector field of interest, or equivalently, if we assume that $c - b = 0$, then the resulting estimated motion vector field is *purely* irrotational. With this fact in mind, given an arbitrary affine motion vector field, we can proceed by first estimating the irrotational component according to the projection-based approach described above. The images can then be warped according to this estimated vector field, and the resulting pair of images will then be *known* to be related by a vector field that is a combination of translational and purely rotational components. While the rotation can not be estimated using a global application of the projection-based method, it is possible to estimate rotation by applying the method locally in smaller windows of the image. It is true that in a window of fixed size, as we move away from the center of rotation, the curl component becomes increasingly small. Therefore, the component of pure rotation in a window away from the center of rotation is measured effectively as a translation. Combining these local estimates with the knowledge that the underlying motion field is *purely* rotational with an unknown center of rotation (the translational component), the curl component of the overall global vector field can very likely be accurately estimated as well. The computational complexity of the overall projection-based method process is, of course, worsened if this additional rotational motion estimation is in fact carried out. We leave further analysis of this problem for future research.

In the present framework, in order to generate estimates for all of the affine parameters, we assume that $c - b = \rho$ where ρ is some known curl value, typically set to zero.

In closing this section, it is also worth observing that we need at least two projection angles to determine the shift vector $\vec{v}_0$ and at least three projection directions to estimate all of the curl-free affine parameters of M. Given an arbitrary affine vector field, we typically employ four projection angles at $\theta = 0, 45, 90,$ and 135 degrees. The choice of these angles can also be optimized as a function of the given image (spatial frequency) content to produce the best possible estimates—this is another interesting topic worthy of future research.

4. Local, Global, and Multiscale Vector Field Estimation

Until now, we have not specified the region of interest where we apply the above estimators. In this section we explain how the previously described models can be applied to the image sequence in a global or local fashion to estimate more diverse vector fields. Then, we show how the estimators are embedded into a hierarchical or multiscale framework to yield improved performance as well as computational efficiency.

In earlier sections, estimators (35) and (11) were applied to an unspecified region in the image where the affine motion model was assumed to characterize the image dynamics. The simplest such region to apply the estimator is the entire image. For this case, we obtain parameters that describe the global motion. When the motion model applies in the global sense, this form of estimation usually produces a very good estimate as often there are thousands of equations used to estimate only six parameters. This model works well to capture image dynamics produced by a moving camera or images of a large rigid object motion where the object fills the camera's field of view.

Another popular approach for estimating more complex vector fields is that of dividing the images into small overlapping or non-overlapping regions. This region-based approach assumes that the simple parametric model characterizes the motion present only in a small region. The more complex vector field $\vec{v}$ is then approximated as a piecewise collection of simpler parametric vector fields. These piecewise vector fields are sometimes forced to satisfy some constraint such as smoothness [13]. The simplest form of local estimation is to find translational motion for small image regions. The translational model of image dynamics $f(x - v_{0x}, y - v_{0y}t)$ is likely to be valid for small

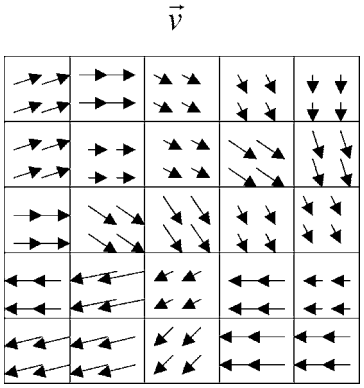

Figure 2. Region based vector field estimation.

spatio-temporal regions in the image sequence. The vector field estimation process begins by estimating the translational motion for each region in the image. Then, these estimates are combined to generate an estimate of the vector field $\vec{v}$. The estimated translational motion for each block represents a sample of the overall vector field. Thus, the dense vector field estimate $\hat{\vec{v}}$ is usually generated by some form of interpolation of these vector field samples. One such form of interpolation is that of replication of the vector samples, where the final vector flow field has regions of constant velocity such as in Fig. 2. This approach is common in video coding where the motions of each block are estimated using a variety of approaches. Some of these approaches include matched filtering, correlation and phase-based methods.

As shown in [23], this local vector field estimation method can be understood as a special case of variable size region-based motion estimation. Multiscale motion estimation attempts to estimate a vector field by estimating the velocity components for variable sized regions at different scales of image resolution. Basically, the multiscale framework estimates a vector field by combining the coarse motion properties in large image regions at low image resolution with the finer motion vector estimates estimated in smaller regions at higher resolution.

To understand the utility of the multiscale framework we first motivate the use of an iterative estimation process. Recall from Sections 2 and 3 the truncation of the Taylor series expansion to the first order used to produce (6) and (21). This approximation assumes a small motion vector $\vec{v}$ (assuming unit time between frames) and is not accurate for regions where the velocity vector $\vec{v}$ is large. The multiscale approach attempts to remedy this inaccuracy by iterating over scale. More specifically, the multiscale approach decomposes the image sequence into a dyadic pyramid of successive sequences of lowpass filtered and downsampled images. At each time (frame), this creates an image pyramid with image sequences at the top having the lowest resolution and size while the original image sequence lies at the bottom. See Fig. 3. The motion vectors describing the dynamics in the downsampled images will necessarily be reduced in magnitude by the downsampling ratio. This reduction in magnitude improves the accuracies of the models (6) and (21) by "shrinking" the magnitude of $\vec{v}$. Furthermore, the lowpass filtering used to construct the image pyramid also serves to regularize the optical flow estimation problem [23].

When the assumption of intensity conservation is violated in an image sequence, the estimates produced by (11) and (35) are biased. The bias results essentially from error in linearizing a nonlinear least squares problem. To mitigate this error the equations of (11) and (35) can be used to generate improved estimates in a Gauss-Newton iterative scheme [4]. The performance of the iterative nonlinear least squares estimators depend on both the convexity of the objective function (sum of the squared image differences) as well as the accuracy of the relative estimate at each iteration.

This Gauss-Newton nonlinear least squares estimation can be combined with the multiscale framework. The iterative multiscale estimation begins by estimating motion in the image sequence at the coarsest scale (the top of the pyramid) denoted by $f^h(x, y, t)$ using (11) or (35) (the superscript of f indicates the level of the pyramid where h is the total height of the pyramid). Because of the image downsampling, the velocity vector x, y components in this image sequence are reduced in magnitude by 2^h. After estimating the vector field $\hat{v}^1$ at the coarsest level, the image sequence at the next finer resolution level of the pyramid $f^{h-1}(x, y, t)$ is warped according to $2\times$ the velocity estimates $\hat{v}^1$ to create a warped image sequence $\check{f}^{h-1}(x, y, t)$ with the estimated coarse image motion removed from the image sequence. Then, the residual motion $\hat{v}^r$ is estimated from this warped image sequence $\check{f}^{h-1}(x, y, t)$ and an updated velocity vector field is generated by $\hat{v}^2 = 2\hat{v}^1 + \hat{v}^r$. This process repeats down the pyramid iterating in a coarse to fine fashion. The multiscale aspect of the iteration serves the additional role of

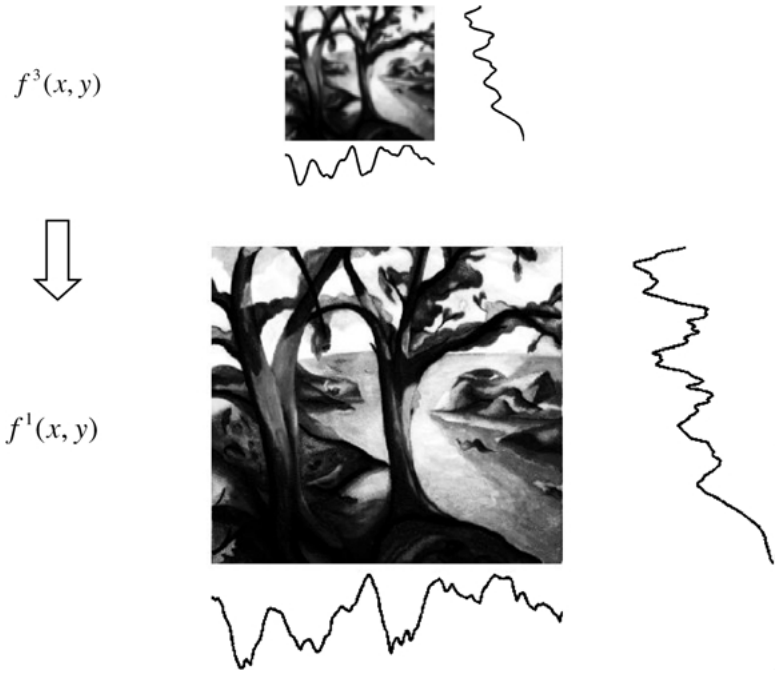

Figure 3. Fake trees image at two pyramid resolutions and the corresponding projections.

reducing computation since the images at the coarsest levels are downsampled (smaller). Thus, the computation time required to warp the image sequences as well as the time required to estimate the residual motions is reduced.

The multiscale iteration can be applied to both the direct and projection based methods for estimating vector fields. The use of multiscale iteration for direct estimation has been shown to produce very accurate results [4]. The multiscale iteration can also be combined with projection based estimation to produce equally good results while realizing significant computational savings. For example, Fig. 3 shows the Fake Trees image at the coarsest resolution ($h = 3$) and at the original image resolution. The corresponding image projections are also shown and are used to estimate global motion. Initially, the global motion parameters are estimated from the projections ($g^h(p, \theta, t)$) of the coarsest image sequence. Again, a warped image sequence at the next pyramid level is generated $\check{f}^{h-1}(x, y, t)$ using the estimates from the projections of the previous level $\vec{v}^1$. A new set of image projections $\check{g}^{h-1}(p, \theta, t)$ are then generated from the warped image sequence $\check{f}^{h-1}(x, y, t)$. As before, this process repeats down the pyramid in a coarse-to-fine fashion.

5. Computational Complexity

In this section we compare the computational complexities of the direct and the projection-based estimators. We will examine the computational cost of estimating the parameters of affine motion between a pair of $L \times L$ images. We are not including any of the cost associated with multiscale estimation as it will pertain to both estimators equally. We distinguish the original estimator from the projection based estimator as being the 2-D and 1-D methods respectively. We assume that N is the number of projections used (typically 3 or 4). For our evaluation of image gradients, we use convolution kernels such that 10 multiplications and additions are required to estimate the 2-D gradient at each pixel and 5 multiplications and additions are required to estimate the derivative at each point in the projection. We obtain the cost for motion estimation as a general cost of solving linear system from [11] where six parameters are estimated in the 2-D case and two are estimated in the 1-D case. Finally, we assume that $N \ll L$ so that the final cost of estimating the 2-D affine parameters from projected motion parameters, is negligible. This leads us to a general overall computational complexity of $\mathcal{O}(46L^2)$ for the direct 2-D estimation and $\mathcal{O}(NL^2 + 9N)$ for the projection-based 1-D estimator (see Table 1).

Table 1. Complexity of gradient-based direct and indirect methods.

Gradient-based estimators	2-D	1-D
Projection	0	NL^2
Gradient	$10L^2$	$5NL$
Motion estimation	$36L^2$	$4NL$
Inverse estimation	0	$36N^2$

We find in practice that using $N = 4$ projection angles to estimate affine motion requires at worst only about 25 percent of the computational time required by the 2-D method, thus realizing significant computational savings. It is important to note that the cost of computing projections, which is the leading term in the complexity of the 1-D method, involves only additions, where as the leading L^2 term in the direct 2-D method involves multiplications. Furthermore, we point out that typically motion estimation methods employ some form of presmoothing of the images prior to motion estimation. We have not included this presmoothing step in our analysis or experiments and we have ignored its computational cost. But we mention here that the computational cost of presmoothing is again significantly lower if this operation is performed on the projections instead of the images.

6. Experiments and Results

Here we present a set of experiments exploring the performance of the direct and indirect methods for estimating affine motion. We begin with experiments estimating global affine vector fields for a set of images in both a non-iterative and multiscale iterative framework. Then, we compare the direct and indirect estimation of affine vector fields using local estimation methods. For our experiments, we use a combination of well-known benchmark image sequences as well as our own generated image sequences.

6.1. Error Measures and Test Image Sequences

Following [3] we measured mean angular error between the correct motion vectors V_c in space-time and the estimated motion vector in space-time V_e. Each space-time velocity vector has the form $V(x, y) = (v_1(x, y), v_2(x, y), 1)^T$ where v_1, v_2 are the velocities in the x-y directions. Thus the motion vectors have unit length in the time dimension. The mean angular error between V_c and V_e is measured by:

$$\overline{\psi_{ang}} = \frac{1}{L^2} \sum_{x,y} \cos^{-1}(V_c(x, y) \cdot V_e(x, y)) \qquad (37)$$

where the sum is taken over all L^2 pixels of interest. To gather more information about the motion estimation methods we also compute the mean magnitude error as:

$$\overline{\psi_{mag}} = \frac{1}{L^2} \sum_{x,y} \|\vec{v}_c(x, y) - \vec{v}_e(x, y)\|_2 \qquad (38)$$

Again, this represents the average magnitude of the error vector over all pixels in the image.

In our experiments, we evaluate the performance of our projection based estimator for both well-known image sequences and also our own synthetic image sequences. To generate a synthetic image sequence, we warp an individual image according to the affine transformation model of (1) to create an image pair. The second image in the pair is a linearly interpolated version of the reference image, where the interpolation is based on a known motion vector field. We then estimate this vector field from the image pair. The images we used to generate synthetic image sequences are shown in Fig. 4.

1. *Forest.* Picture of a forest contains similar image statistics to those of a natural scene with rich textures. The image is 301 × 447 pixels.

Figure 4. Experimental test images: Forest and Lab images.

2. *Lab.* Picture from a webcam at the researchers' office. The webcam was rotated about 45° so as to create an image in which the majority of image texture is not aligned at 0° and 90°. The image is 240×320 pixels.

In addition to our own synthetic image sequences, we follow the papers of [3, 16] and measure performance on a well known set of benchmark image sequences from [3]. While these image sequences contain many frames, we limit the image sequences to only five frames. In practice, this represents a reasonable number of frames as often in real image sequences the vector field $\vec{v}$ might only remain static for a short period of time. The image sequences that we use are the following:

1. *Diverging Tree.* The image sequence imitates a camera zooming into scene creating a divergent motion vector field.
2. *Translating Tree.* The image sequence contains translational motion which is very close to global translation arising from camera motion in the x-direction.

Both of these image sequences are based on the image shown in Fig. 10. We apply both the global and local estimators to these benchmark sequences.

For each set of global estimation experiments we added zero-mean Gaussian noise to produce the specified image signal to noise ratio (SNR).[4] The motion vector fields were estimated from these noisy image sequences and the corresponding error measures for the estimates were calculated. For each experiment, we repeated the estimation process 100 times at each SNR and averaged the error values. We evaluate the performance of the local estimation methods without any additive noise so as to compare the results with those of [3].

6.2. *Global Vector Field Estimation*

We begin our experimental performance analysis by estimating global affine vector fields described by the affine motion model of (2). As mentioned in Section 3, the rotational component of the affine vector field cannot be directly estimated using the global projection-based estimator. Therefore, we first examine the performance of the method in estimating affine vector fields constrained to have no rotational component, and we compare the results to the performance of the direct 2-D method.[5] We then extend the experiments to include estimation of the general affine model to understand the indirect estimator's performance in the presence of image rotation. For the projection-based estimation, we use four projection angles of 0°, 45°, 90° and 135° in each experiment.

We initially examine the performance of the projection based global estimator on the benchmark Translating and Diverging tree sequences, which contain no rotational component. The plots of Figs. 5 and 6 show the performance of the 1-D and 2-D methods using

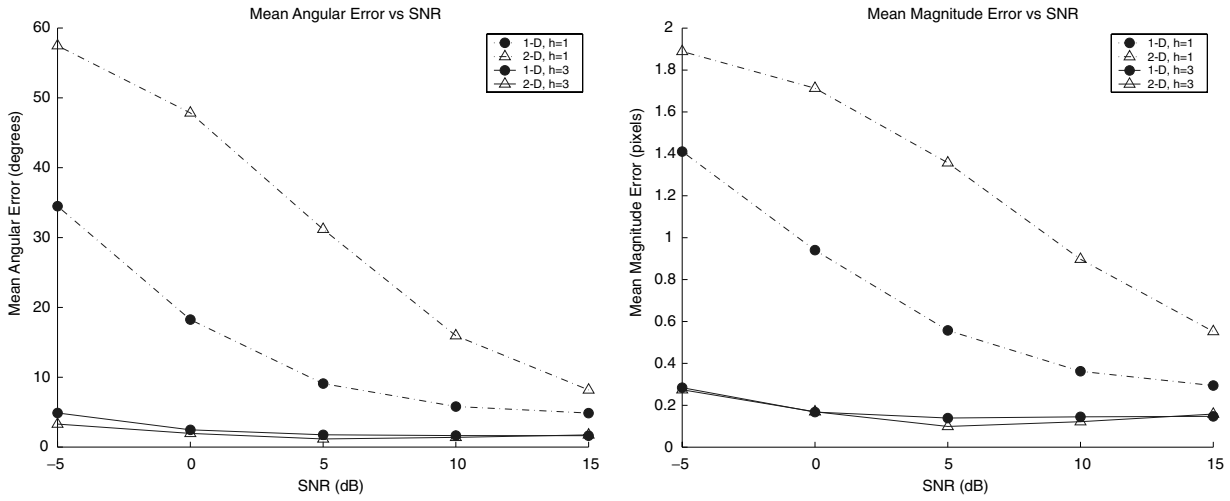

Figure 5. Mean angular and magnitude error for the translating tree sequence.

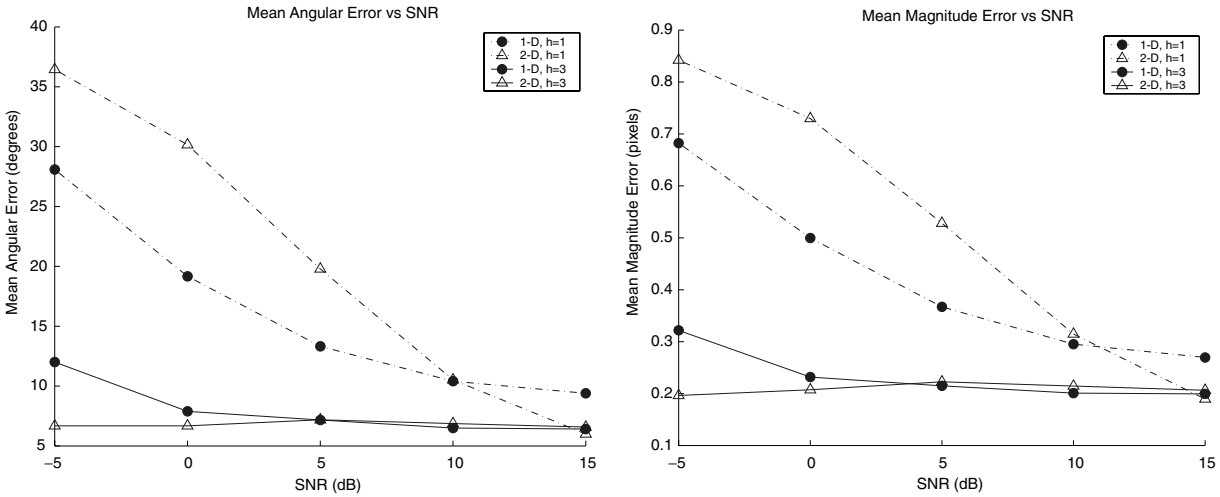

Figure 6. Mean angular and magnitude error for the diverging tree sequence.

no multiscale iteration ($h = 1$, dashed lines) and for a multiscale pyramid of height $h = 3$ (solid lines). The triangles indicate the error of the 2-D method and the circles indicate the error of the 1-D projection based estimator. We follow this graphical format for all of the experiments on global affine vector field estimation.

From Figs. 5 and 6, we see that the projection-based estimator *outperforms* the direct 2-D method when the method is not iterated in multiscale, but the difference in performance shrinks as the SNR improves. For both image sequences, when motion is estimated using multiscale iteration, the performance of the direct and projection based estimators are essentially equivalent. Only for very poor SNR in the case of the Diverging Tree sequence (Fig. 6) do we see a small performance difference between the 1-D and 2-D methods.

To evaluate the performance of the projection-based estimator more systematically using simulated motion, we continue our experimentation using our synthetic image sequences. Figure 7 shows the performance of both the 2-D and 1-D methods in estimating the global affine vector field with parameters $M = \left[\begin{smallmatrix} .05 & .01 \\ .01 & .06 \end{smallmatrix}\right]$ and $\vec{v}_0 = \left[\begin{smallmatrix} .5 \\ .5 \end{smallmatrix}\right]$ applied to the Forest image.

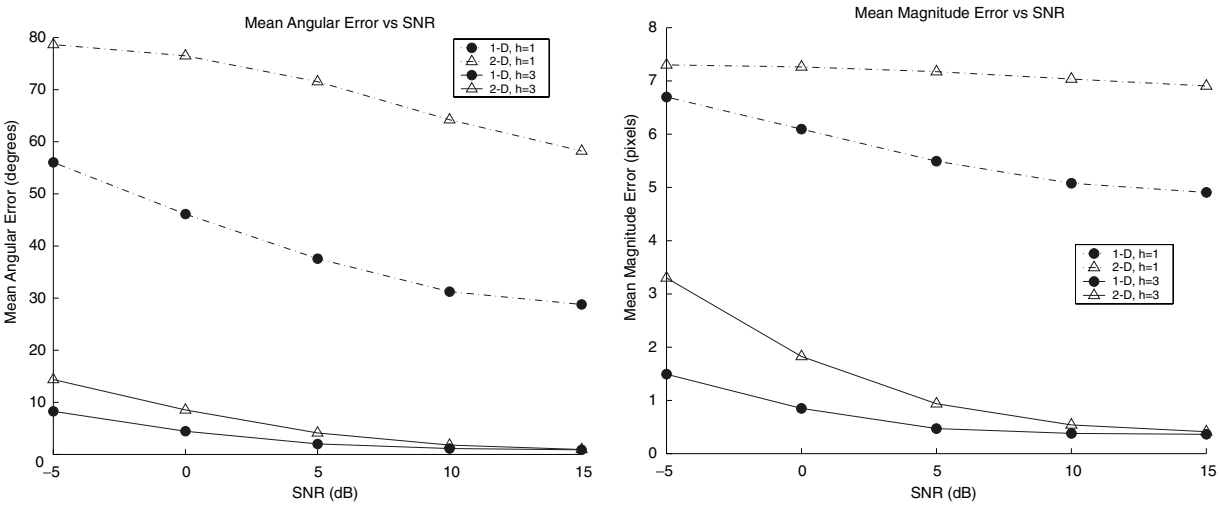

Figure 7. Mean angular and magnitude error for the Forest image with constrained motion.

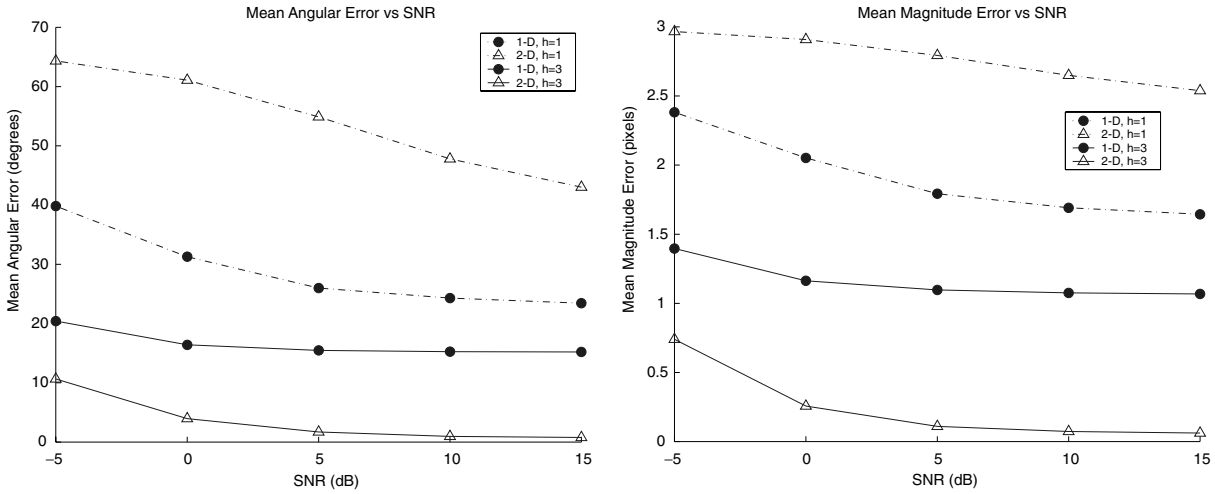

Figure 8. Mean angular and magnitude errors for the lab image with rotation.

As a point of reference, for a particular realization of noise at SNR of 5 dB, the 1-D estimator using multiscale ($h = 3$) iteration produces estimates of

$$\hat{M} = \begin{bmatrix} .0484 & .0079 \\ .0079 & .0382 \end{bmatrix}$$

and

$$\hat{v}_0 = \begin{bmatrix} .3223 \\ .4986 \end{bmatrix}$$

which corresponds to mean angular error of 1.8 degrees and a mean magnitude error of 0.39 pixels. Using the same data, the 2-D estimator produces

$$\hat{M} = \begin{bmatrix} .0471 & .0080 \\ .0080 & .0339 \end{bmatrix}$$

and

$$\hat{v}_0 = \begin{bmatrix} .3885 \\ .1760 \end{bmatrix}$$

which corresponds to a mean angular error of 3.19 degrees and a mean magnitude error of 0.68 pixels.

Again, we see the non-iterative projection based estimator outperforming the direct 2-D estimator. Using the multiscale iteration, the 1-D projection based estimator continues to outperform the 2-D method. As the SNR improves, both methods seem to converge to similar performance. We present these results as a representative sample of the many experiments we carried out using other irrotational affine vector fields as well as different reference images.

To analyze the performance for the case of general affine motion, we estimate image dynamics for a vector field containing nonzero curl. Figure 8 shows the errors in estimating a vector field applied to the Lab image with affine parameters

$$M = \begin{bmatrix} -.01 & -.01 \\ -.03 & .02 \end{bmatrix}$$

and

$$\vec{v}_0 = \begin{bmatrix} .5 \\ .5 \end{bmatrix}.$$

As the plot indicates, without using multiscale iteration, the projection-based 1-D estimator again outperforms the direct estimator. However, employing a multiscale pyramid of height $h = 3$, the 2-D method clearly produces better estimates of the vector field. While the multiscale iteration does improve the projection based estimates, the iterations only improve the estimate of the irrotational component of motion. For example, at a SNR of 5 dB and multiscale height $h = 3$, the projection based method produces affine parameter estimates of

$$\hat{M} = \begin{bmatrix} -.0093 & -.0238 \\ -.0238 & .0178 \end{bmatrix}$$

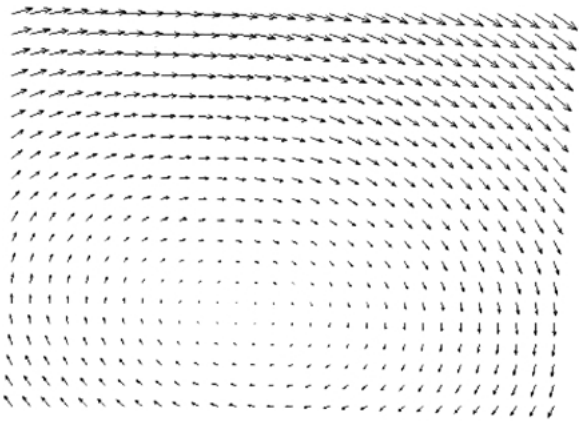

Figure 9. Residual velocity vector field for projection-based estimation of general affine vector field.

and

$$\hat{v}_0 = \begin{bmatrix} -.6807 \\ .0944 \end{bmatrix}.$$

The residual motion vector field $\vec{v} - \hat{\vec{v}}$ is shown in Fig. 9. The figure shows that the residual motion not captured by the projection based estimator is primarily the rotational component of affine motion, though the translational component seems to have been affected as well. By contrast, the 2-D estimator for the same image pair produces the estimates

$$\hat{M} = \begin{bmatrix} -.0106 & -.0097 \\ -.0294 & .0188 \end{bmatrix}$$

and

$$\hat{v}_0 = \begin{bmatrix} -.5291 \\ .4231 \end{bmatrix}.$$

These experiments indicate that when the motion is constrained such that there is no image rotation, the 1-D method performs just as well if not better than the 2-D method for global affine motion estimation. The notion that the 1-D method can perform better than the 2-D method in some circumstances deserves a systematic and careful future study. The previous figures also show that the multiscale iteration can provide substantial improvements in performance for both the non-iterative 1-D and 2-D estimators.

6.3. Local Vector Field Estimation

Finally, we present experiments with the use of projections for estimating local motion in a block-based scheme as outlined in Section 4. As mentioned earlier, application of the direct gradient-based translational estimation of Section 2 to small blocks in an image sequence was first introduced by Lucas and Kanade [17]. Here we compare the performance of an indirect block based translational estimation scheme with the direct method of [17]. The direct gradient method consistently performs well as shown in most optical flow estimation survey papers such as [3] and [16]. We will show that this performance also extends to the projection-based method, while significantly improving the computational efficiency.

As indicated in Section 4, both the direct and indirect techniques require choosing a set of operating parameters, ultimately affecting estimator performance. For instance, both methods initially subdivide the image into blocks for which a motion vector is estimated. The choice of block sizes plays a critical role in determining both the accuracy and the speed of the techniques. Furthermore, depending on a desired density of the motion vector field, the size of the blocks affects the amount of block overlap. Both methods must choose a number of images to use in calculating one motion vector field. Finally, each of the projection-based approaches requires a pair of projection angles.

To improve the performance of the block based estimators, we apply a weighting vector to the least squares estimator which weights the pixels at the center of the block more than the pixels at the periphery. We denote this weighting function $w(x, y)$ for the direct estimator and $w(p)$ for the indirect estimator. Applying this weighting function to larger blocks will maximize accuracy while minimizing the aperture problem. Basically, the weighting function forces the estimator to estimate motion primarily from the pixels at the center of the block but also allows pixels at the periphery of the block to influence the estimate slightly. To simplify the characterization of the weighting function, we use Gaussian function $w(p) = e^{-p^2/\gamma}$. The weighting function is parameterized by γ, or the variance of the Gaussian function.

To directly compare the 1-D block based estimator with the 2-D block based method in a fashion similar to [3], we estimated the affine vector fields for Translating and Diverging tree sequences using a block size of 30×30 pixels which appears to produce the overall

Table 2. Results for all three sequence.

Estimation method	1-D Tran	2-D Tran	1-D Div	2-D Div
Mean angular error (degrees)	11.385	14.108	5.888	6.112
Standard deviation	0.7064	0.6470	0.3325	0.3361
Mean magnitude error (pixels)	0.574	0.778	0.153	0.169
Standard deviation	0.0269	0.0231	0.0094	0.0110
Cpu time (s)	1.920	23.880	1.930	24.030

best results for both methods. We then used both estimators on each sequence using 15 frames and tabulated the results in Table 2. The same table also includes the computation time required to estimate the vector fields.

From Table 2, we observe that the accuracy of the 1-D and 2-D methods appear to be statistically equivalent. The computational complexity, however, is dramatically reduced in the projection-based approaches. The 1-D method's total computation time was on average about 90 percent better than the 2-D counterpart. As a visual example, Fig. 10 shows the estimated motion vector fields for the Diverging Tree image sequence overlaid atop one image of the sequence. Note that the motion vector fields are visually quite similar.

7. Conclusions and Future Work

In the paper we introduced a unified framework for the estimation of affine motion using projections. Previous attempts at the same were mostly ad-hoc and, most importantly, did not address the question of relative performance between the direct 2-D methods and the proposed 1-D approaches. Here we have shown that projection-based methods offer a computationally very attractive alternative to the direct methods, while in most cases maintaining or even improving the level of accuracy. The idea that projection-based methods can often display improved performance is a theoretically intriguing one and deserves careful study in the future. We have also shown that the projection-based method can be combined with a multiscale iterative framework to provide further accuracy in motion estimation while minimizing computation time.

These results suggest much room for future research in the area of estimating motion using projections. For instance, the gradient-based method of estimating is only one method of many for estimating motion using projections. Phase-based methods are another possibility that should be explored. Improved performance may also be realized by using more sophisticated statistically robust methods in place of the least squares approach presented in this paper.

Finally, some of our preliminary experimentation has indicated that the choice of projection angles plays a fundamentally important role in the performance of any projection based motion estimation method. Adaptively identifying the optimal set of projection angles, as a function of the given images, for best estimator performance remains an open question.

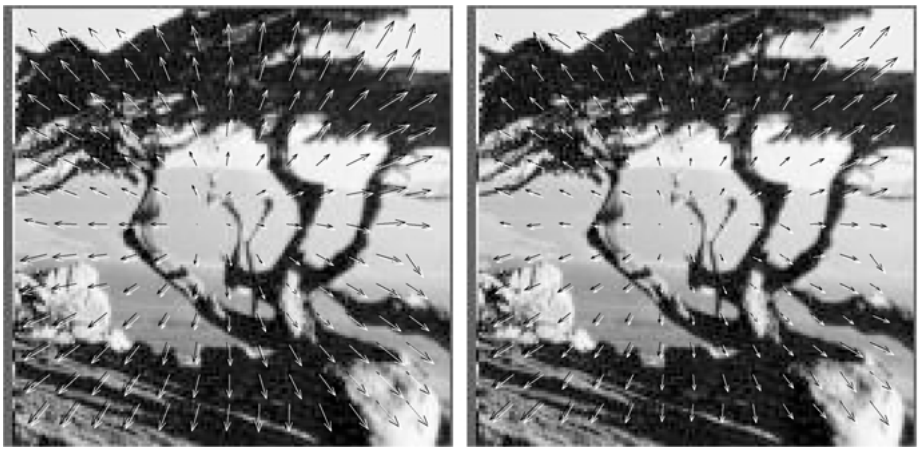

Figure 10. Motion vector field from 2-D (left) and 1-D (right) methods.

Appendix A. Calculating Derivatives in Image Projections

Here we will introduce the intuitive reasoning for applying a weighting to the projection images prior to calculating derivatives used in estimating projected motion. We shall explain how this weighting acts as a modification of the spatial derivative operator. Because the image under projection is defined over a rectangular region of samples, different points in the projection are generated by integrating over lines of varying length. In terms of image pixels, this means that different points in the projection integrate different numbers of pixels in the original image. Thus, a rectangular constant valued image on $[-w/2, w/2] \times [-h/2, h/2]$ would not appear flat in the projection image but rather as a piecewise linear function (see Fig. 11) given by

$$\begin{aligned}
&\mathcal{R}[f(x, y) = c] \\
&\quad = \int_{-h/2}^{h/2} \int_{-w/2}^{w/2} c\delta(p - x\cos(\theta) - y\sin(\theta))\, dx\, dy \\
&\quad = \int_{S^-(p,\theta)}^{S^+(p,\theta)} c\, ds \\
&\quad = S^+(p, \theta) - S^-(p, \theta) = S(\theta) = g(p, \theta)
\end{aligned} \tag{39}$$

where

$$\begin{aligned}
&S^+(p, \theta) \\
&\quad = \min\left[p\cot\theta + \frac{w}{2\sin\theta}, -p\tan\theta + \frac{h}{2\cos\theta} \right]
\end{aligned} \tag{40}$$

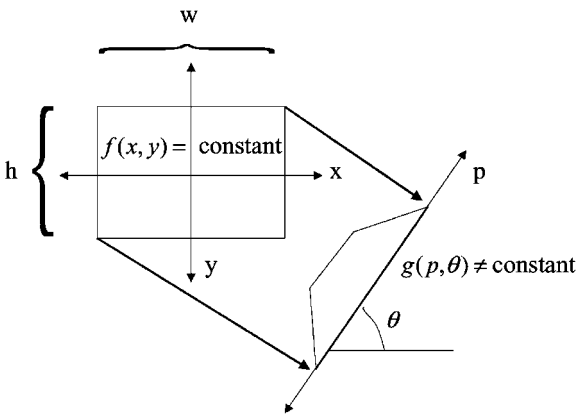

Figure 11. Projection of a constant image

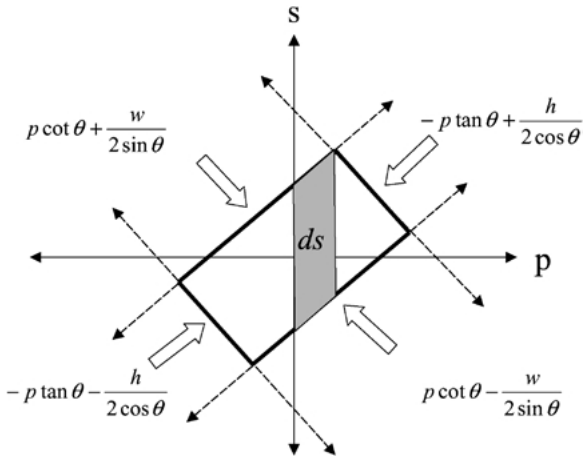

Figure 12. Integration region.

$$\begin{aligned}
&S^-(p, \theta) \\
&\quad = \max\left[p\cot\theta - \frac{w}{2\sin\theta}, -p\tan\theta - \frac{h}{2\cos\theta} \right]
\end{aligned} \tag{41}$$

Here, the functions S^+, S^- come from the edges of the rectangular image region. See Fig. 12. Thus, $g(p, \theta)$ is a piecewise linear function whose derivative will not be zero. Of course, projections at 0 and 90 degrees do not suffer from this anomaly. We propose to normalize the projections such that the projection of a constant image will produce a constant 1-D function. To accomplish this we use a normalized Radon transform of the form

$$\begin{aligned}
&\tilde{g}(p, \theta) \\
&\quad = \tilde{\mathcal{R}}_\theta[f(x, y)] \\
&\quad = \frac{\int\int f(x, y)\delta(p - x\cos\theta - y\sin\theta)\, dx\, dy}{\int\int \delta(p - x\cos\theta - y\sin\theta)\, dx\, dy}
\end{aligned} \tag{42}$$

After computing the normalized Radon transform, we compute the derivatives of the projection at a specific angle θ by

$$g_p(p, \theta) = \tilde{g}(p, \theta) * K(p) \tag{43}$$

where K represents the derivative convolution kernel. This will ensure that the proper spatial derivatives are calculated in the projection based motion estimators.

Appendix B. Linearized Projected Affine Motion

In this section, we derive the Maclaurin series approximation of the exact form of the projected motion function $u(p, \theta)$ for affine motion. From (18) we see that the exact form of the affine motion under projection is

$$u_{exact}(p, \theta) = \vec{v}_0^T \vec{w} + \left(1 - \frac{|\det(J)|}{\|J^T w\|_2}\right) p \tag{44}$$

We show how the coefficient of the second term in the above expression can be linearized by expanding it in a first order Maclaurin series. To begin, let us define

$$\alpha_{exact}(J) = 1 - \frac{|\det(J)|}{\|J^T w\|_2}. \tag{45}$$

Next, we rewrite (45) as a function of the four affine parameters as follows

$$\begin{aligned}
&\alpha_{exact}(J) \\
&\quad = \alpha_{exact}(a, b, c, d) \\
&\quad = 1 - \frac{|1 - a - d + ad - bc|}{[((1-d)w_1 + cw_2)^2 + (bw_1 + (1-a)w_2)^2]^{1/2}}
\end{aligned} \tag{46}$$

The first order Maclaurin series of $\alpha(a, b, c, d)$ will have the form

$$\begin{aligned}
\alpha_{exact}(a, b, c, d) = \alpha(0, 0, 0, 0) + a\alpha_a(0) + b\alpha_b(0) \\
+ c\alpha_c(0) + d\alpha_d(0)
\end{aligned} \tag{47}$$

To simplify the derivation, we write

$$\alpha_{exact}(a, b, c, d) = 1 - \beta(a, b, c, d)\zeta^{-1/2}(a, b, c, d) \tag{48}$$

where

$$\beta(a, b, c, d) = |1 - a - d + ad - bc| \tag{49}$$

and

$$\begin{aligned}
\zeta(a, b, c, d) = ((1-d)w_1 + cw_2)^2 \\
+ (bw_1 + (1-a)w_2)^2
\end{aligned} \tag{50}$$

Thus, from the chain rule we see that the partial derivatives of α_{exact} will have the form

$$\begin{aligned}
\alpha_x &= -\left[\frac{\partial \beta}{\partial x}(\zeta^{-1/2}) - \left(\beta \frac{1}{2}\zeta^{-3/2}\right)\frac{\partial \zeta}{\partial x}\right] \\
&= \left[\left(\beta \frac{1}{2}\zeta^{-3/2}\right)\frac{\partial \zeta}{\partial x} - \frac{\partial \beta}{\partial x}(\zeta^{-1/2})\right].
\end{aligned} \tag{51}$$

Next, we note that $\alpha_{exact}(0) = 0$, $\zeta(0) = 1$ and $\beta(0) = 1$.

We now compute the partial derivatives of β evaluated at 0.

$$\begin{aligned}
\beta_a(0) &= -1 \\
\beta_b(0) &= 0 \\
\beta_c(0) &= 0 \\
\beta_d(0) &= -1
\end{aligned}$$

Likewise, we now evaluate the partial derivatives of ζ.

$$\begin{aligned}
\zeta_a(0) &= -2w_2^2 \\
\zeta_b(0) &= 2w_1 w_2 \\
\zeta_c(0) &= 2w_1 w_2 \\
\zeta_d(0) &= -2w_1^2
\end{aligned}$$

Finally, we see that the partial derivatives of α_{exact} are

$$\begin{aligned}
\alpha_a(0) &= 1 - w_2^2 = w_1^2 \\
\alpha_b(0) &= w_1 w_2 \\
\alpha_c(0) &= w_1 w_2 \\
\alpha_d(0) &= 1 - w_1^2 = w_2^2
\end{aligned}$$

Combining these calculations, we obtain the following linearization of α_{exact}:

$$\begin{aligned}
\alpha_{exact}(a, b, c, d) &\approx a w_1^2 + b w_1 w_2 + c w_1 w_2 + d w_2^2 \\
&= \vec{w}^T M \vec{w}
\end{aligned} \tag{52}$$

This is the same form of projected affine motion obtained using the PMI assumption, discussed in (17).

Acknowledgments

We would like to thank Dr. Hai Tao of UC Santa Cruz Computer Engineering for suggesting the multiscale extensions, and Dr. Michael Elad of Stanford University for reading and commenting on the manuscript.

Notes

1. As will be elaborated later, the the curl of the motion field, however, is not directly measurable in the projections.
2. We note here that $g(p, \theta, t)$ is the Radon transform of $f(x, y, t)$ for each fixed t.
3. We note that the subscript 1 on $\mathbf{H}_1$ and $\mathbf{\Psi}_1$ refers to the 1-D nature of the derived estimation problem.
4. Signal to noise ratio (SNR) is defined as $10 \log_{10} \frac{\sigma_c^2}{\sigma^2}$ where σ_c^2 and σ^2 are the variances of a clean frame and the noise respectively.
5. In the interest of fairness, the 2-D method employed in estimating these irrotational vector fields employed *constrained* least squares with the constraint that $c - b = 0$. The plots of Figs. 5–7 reflect the use of this constraint in the 2-D case.

References

1. A. Akutsu and Y. Tonomura, "Video tomography: An efficient method for camerawork extraction and motion analysis," *Transactions of the Institute of Electronics, Information, and Communications Engineers*, Vol. J79D-II, No. 5, pp. 675–686, 1996.
2. S. Alliney and C. Morandi, "Digital image registration using projections," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 8, No. 2, pp. 222–233, 1986.
3. J.L. Barron, D.J. Fleet, S.S. Beauchemin, and T.A. Burkitt, "Performance of optical flow techniques," *CVPR*, Vol. 92, pp. 236–242, 1992.
4. J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani, "Hierachical model-based motion estimation," in *Proceedings European Conference on Computer Vision*, 1992, pp. 237–252.
5. C. Bergeron and E. Dubois, "Gradient-based algorithms for block-oriented MAP estimation of motion and application to motion-compensated temporal interpolation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 1, pp. 72–85, 1991.
6. M.J. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *Computer Vision and Image Understanding*, Vol. 63, pp. 75–104, 1996.
7. L.G. Brown, "A survey of image registration techniques," *ACM Computing Surveys*, Vol. 24, No. 4, pp. 325–376, 1992.
8. S.C. Cain, M.M. Hayat, and E.E. Armstrong, "Projection-based image registration in the presence of fixed-pattern noise," *IEEE Transactions on Image Processing*, Vol. 10, No. 12, pp. 1860–1872, 2001.
9. F. Coudert, J. Benois-Pineau, and D. Barba, "Dominant motion estimation and video partitioning with a 1-D signal approach," in *SPIE Conference on Multimedia Storage and Archiving Systems III*, Vol. 3527, 1998, pp. 283–294.
10. S. R. Deans, *The Radon Transform and Some of its Applications*, John Wiley and Sons: New York, 1983.
11. M.T. Heath, *Scientific Computing: An Introductory Survey*, McGraw-Hill: New York, 2002.
12. B.K.P. Horn, *Robot Vision*, MIT Press: Cambridge, 1986.
13. A. Jepson and M. Black, "Mixture models for optical flow computation," in *Proceedings Computer Vision and Pattern Recognition*, June 1993, pp. 760–761.
14. S.M. Kay, *Fundamentals of Signal Processing: Estimation Theory*, Prentice Hall: Englewood Cliff, NJ, 1993.
15. Joon-Seek Kim and Rae-Hong Park, "A fast feature-based block matching algorithm using integral projections," *IEEE Journal on Selected Areas in Communications*, Vol. 10, No. 5, pp. 968–971, 1992.
16. Hongche Liu, Tsai-Hong Hong, M. Herman, T. Camus, and R. Chellappa, "Accuracy vs. efficiency trade-offs in optical flow algorithms," *Computer Vision and Image Understanding*, Vol. 72, pp. 271–286, 1998.
17. B.D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *DARPA81*, 1981, pp. 121–130.
18. P. Milanfar, "Projection-based, frequency-domain estimation of superimposed translational motions," *Journal of the Optical Society of America*, Vol. 13, No. 11, pp. 2151–2162, 1996.
19. P. Milanfar, "A model of the effect of image motion in the Radon transform domain," *IEEE Transactions on Image Processing*, Vol. 8, No. 9, pp. 1276–1281, 1999.
20. S.A. Rajala, A.M. Riddle, and W.E. Snyder, "Application of the one-dimensional Fourier transform for tracking moving objects in noisy environments," *Computer Vision, Graphics, and Image Processing*, Vol. 21, pp. 280–293, 1983.
21. D. Robinson and P. Milanfar, "Accuracy and efficiency tradeoffs in using projections for motion estimation," in *Proceedings of the 35th Asilomar Conference on Signals, Systems, and Computers*, November 2001.
22. S.A. Seyedin, "Motion estimation using the Radon transform in dynamic scenes," in *Proceedings of the International Society for Optical Engineering*, 1995, Vol. 2501, pp. 1337–1348.
23. C. Stiller and J. Konrad, "Estimating motion in image sequences," *IEEE Signal Processing Magazine*, Vol. 16, pp. 70–91, 1999.
24. T. Tsuboi, A. Masubuchi, and S. Hirai, "Video-frame rate detection of position and orientation of planar motion objects using one-sided Radon transform," in *Proceedings IEEE Conference of Robotics and Automation*, April 2001, Vol. 2, pp. 1233–1238.
25. Chengjie Tu, T.D. Tran, J.L. Prince, and P. Topiwala, "Projection-based block matching motion estimation," in *Proc. SPIE Applications of Digital Image Processing XXIII*, August 2000, pp. 374–384.
26. Jiangsheng You, Weignou Lu, Jian Li, Gene Gindi, and Zhengrong Liang, "Image matching for translation, rotation, and uniform scaling by the Radon transform," in *Proceedings International Conference on Image Processing*, 1998, Vol. 1, pp. 847–851.

Dirk Robinson is currently a graduate student in the Electrical Engineering Department at the University of California at Santa Cruz, where he is pursuing his Ph.D. degree. He obtained his B.S. degree in Electrical Engineering from Calvin College in June 2000.

Peyman Milanfar received the B.S. degree in engineering mathematics from the University of California, Berkeley, in 1988, and the S.M., E.E., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1990, 1992, and 1993, respectively. From 1993 to 1994, he was a Member of Technical Staff at Alphatech, Inc., Burlington, MA. From 1994 to 1999, he was a Senior Research Engineer at SRI International, Menlo Park, CA. He is currently Associate Professor of electrical engineering at the University of California, Santa Cruz. He was a Consulting Assistant Professor of computer science at Stanford University from 1998–2000. His technical interests are in statistical and numerical methods for signal and image processing, and more generally inverse problems. Dr. Milanfar was awarded a National Science Foundation CAREER grant in January 2000. He was associate editor for the IEEE Signal Processing Letters from 1999 to 2002, and is a Senior member of the IEEE.