Hierarchical Model-Based Motion Estimation

James R. Bergen, P. Anandan, Keith J. Hanna, and Rajesh Hingorani

David Sarnoff Research Center, Princeton NJ 08544, USA

Abstract. This paper describes a hierarchical estimation framework for the computation of diverse representations of motion information. The key features of the resulting framework (or family of algorithms) are a *global model* that constrains the overall structure of the motion estimated, a *local model* that is used in the estimation process, and a coarse-fine refinement strategy. Four specific motion models: affine flow, planar surface flow, rigid body motion, and general optical flow, are described along with their application to specific examples.

1 Introduction

A large body of work in computer vision over the last 10 or 15 years has been concerned with the extraction of motion information from image sequences. The motivation of this work is actually quite diverse, with intended applications ranging from data compression to pattern recognition (alignment strategies) to robotics and vehicle navigation. In tandem with this diversity of motivation is a diversity of representation of motion information: from optical flow, to affine or other parametric transformations, to 3-d egomotion plus range or other structure. The purpose of this paper is to describe a common framework within which all of these computations can be represented.

This unification is possible because all of these problems can be viewed from the perspective of image registration. That is, given an image sequence, compute a representation of motion that best aligns pixels in one frame of the sequence with those in the next. The differences among the various approaches mentioned above can then be expressed as different parametric representations of the alignment process. In all cases the function minimized is the same; the difference lies in the fact that it is minimized with respect to different parameters.

The key features of the resulting framework (or family of algorithms) are a *global model* that constrains the overall structure of the motion estimated, a *local model* that is used in the estimation process [1], and a coarse-fine refinement strategy. An example of a global model is the rigidity constraint; an example of a local model is that displacement is constant over a patch. Coarse-fine refinement or hierarchical estimation is included in this framework for reasons that go well beyond the conventional ones of computational efficiency. Its utility derives from the nature of the objective function common to the various motion models.

1.1 Hierarchical estimation

Hierarchical approaches have been used by various researchers e.g., see [2, 10, 11, 22, 19]). More recently, a theoretical analysis of hierarchical motion estimation was described in

[1] Because this model will be used in a multiresolution data structure, it is "local" in a slightly unconventional sense that will be discussed below.

[8] and the advantages of using parametric models within such a framework have also been discussed in [5].

Arguments for use of hierarchical (i.e. pyramid based) estimation techniques for motion estimation have usually focused on issues of computational efficiency. A matching process that must accommodate large displacements can be very expensive to compute. Simple intuition suggests that if large displacements can be computed using low resolution image information great savings in computation will be achieved. Higher resolution information can then be used to improve the accuracy of displacement estimation by incrementally estimating small displacements (see, for example, [2]). However, it can also be argued that it is not only *efficient* to ignore high resolution image information when computing large displacements, in a sense it is *necessary* to do so. This is because of aliasing of high spatial frequency components undergoing large motion. Aliasing is the source of false matches in correspondence solutions or (equivalently) local minima in the objective function used for minimization. Minimization or matching in a multiresolution framework helps to eliminate problems of this type. Another way of expressing this is to say that many sources of non-convexity that complicate the matching process are not stable with respect to scale.

With only a few exceptions ([5, 9]), much of this work has concentrated on using a small family of "generic" motion models within the hierarchical estimation framework. Such models involve the use of some type of a smoothness constraint (sometimes allowing for discontinuities) to constrain the estimation process at image locations containing little or no image structure. However, as noted above, the arguments for use of a multiresolution, hierarchical approach apply equally to more structured models of image motion.

In this paper, we describe a variety of motion models used within the same hierarchical framework. These models provide powerful constraints on the estimation process and their use within the hierarchical estimation framework leads to increased accuracy, robustness and efficiency. We outline the implementation of four new models and present results using real images.

1.2 Motion Models

Because optical flow computation is an underconstrained problem, *all* motion estimation algorithms involve additional assumptions about the structure of the motion computed. In many cases, however, this assumption is not expressed explicitly as such, rather it is presented as a regularization term in an objective function [14, 16] or described primarily as a computational issue [18, 4, 2, 20].

Previous work involving explicitly model-based motion estimation includes direct methods [17, 21], [13] as well as methods for estimation under restricted conditions [7, 9]. The first class of methods uses a global egomotion constraint while those in the second class of methods rely on parametric motion models within local regions. The description "direct methods" actually applies equally to both types.

With respect to motion models, these algorithms can be divided into three categories: (i) fully parametric, (ii) quasi-parametric, and (iii) non-parametric. Fully parametric models describe the motion of individual pixels within a region in terms of a parametric form. These include affine and quadratic flow fields. Quasi-parametric models involve representing the motion of a pixel as a combination of a parametric component that is valid for the entire region and a local component which varies from pixel to pixel. For instance, the rigid motion model belongs to this class: the egomotion parameters constrain the local flow vector to lie along a specific line, while the local depth value determines the

exact value of the flow vector at each pixel. By non-parametric models, we mean those such as are commonly used in optical flow computation, i.e. those involving the use of some type of a smoothness or uniformity constraint.

A parallel taxonomy of motion models can be constructed by considering local models that constrain the motion in the neighborhood of a pixel and global models that describe the motion over the entire visual field. This distinction becomes especially useful in analyzing hierarchical approaches where the meaning of "local" changes as the computation moves through the multiresolution hierarchy. In this scheme fully parametric models are global models, non-parametric models such as smoothness or uniformity of displacement are local models, and quasi-parametric models involve both a global and a local model. The reason for describing motion models in this way is that it clarifies the relationship between different approaches and allows consideration of the range of possibilities in choosing a model appropriate to a given situation. Purely global (or fully parametric) models in essence trivially imply a local model so no choice is possible. However, in the case of quasi- or non-parametric models, the local model can be more or less complex. Also, it makes clear that by varying the size of local neighborhoods, it is possible to move continuously from a partially or purely local model to a purely global one.

The reasons for choosing one model or another are generally quite intuitive, though the exact choice of model is not always easy to make in a rigorous way. In general, parametric models constrain the local motion more strongly than the less parametric ones. A small number of parameters (e.g., six in the case of affine flow) are sufficient to completely specify the flow vector at every point within their region of applicability. However, they tend to be applicable only within local regions, and in many cases are approximations to the actual flow field within those regions (although they may be very good approximations). From the point of view of motion estimation, such models allow the precise estimation of motion at locations containing no image structure, provided the region contains at least a few locations with significant image structure.

Quasi-parametric models constrain the flow field less, but nevertheless constrain it to some degree. For instance, for rigidly moving objects under perspective projection, the rigid motion parameters (same as the egomotion parameters in the case of observer motion), constrain the flow vector at each point to lie along a line in the velocity space. One dimensional image structure (e.g., an edge) is generally sufficient to precisely estimate the motion of that point. These models tend to be applicable over a wide region in the image, perhaps even the entire image. If the local structure of the scene can be further parametrized (e.g., planar surfaces under rigid motion), the model becomes fully parametric within the region.

Non-parametric models require local image structure that is two-dimensional (e.g., corner points, textured areas). However, with the use of a smoothness constraint it is usually possible to "fill-in" where there is inadequate local information. The estimation process is typically more computationally expensive than the other two cases. These models are more generally applicable (not requiring parametrizable scene structure or motion) than the other two classes.

1.3 Paper Organization

The remainder of the paper consists of an overview of the hierarchical motion estimation framework, a description of each of the four models and their application to specific examples, and a discussion of the overall approach and its applications.

2 Hierarchical Motion Estimation

Figure 1 describes the hierarchical motion estimation framework. The basic components of this framework are: (i) pyramid construction, (ii) motion estimation, (iii) image warping, and (iv) coarse-to-fine refinement.

There are a number of ways to construct the image pyramids. Our implementation uses the Laplacian pyramid described in [6], which involves simple local computations and provides the necessary spatial-frequency decomposition.

The motion estimator varies according to the model. In all cases, however, the estimation process involves SSD minimization, but instead of performing a discrete search (such as in [3]), Gauss-Newton minimization is employed in a refinement process. The basic assumption behind SSD minimization is *intensity constancy.* as applied to the Laplacian pyramid images. Thus,

$$I(\mathbf{x},t) = I(\mathbf{x} - \mathbf{u}(\mathbf{x}), t-1)$$

where $\mathbf{x} = (x,y)$ denotes the spatial image position of a point, I the (Laplacian pyramid) image intensity and $\mathbf{u}(\mathbf{x}) = (u(x,y), v(x,y))$ denotes the image velocity at that point. the SSD error measure for estimating the flow field within a region is:

$$E(\{\mathbf{u}\}) = \sum_{\mathbf{x}} (I(\mathbf{x},t) - I(\mathbf{x} - \mathbf{u}(\mathbf{x}), t-1))^2 \tag{1}$$

where the sum is computed over all the points within the region and $\{\mathbf{u}\}$ is used to denote the entire flow field within that region. In general this error (which is actually the sum of individual errors) is not quadratic in terms of the unknown quantities $\{\mathbf{u}\}$, because of the complex pattern of intensity variations. Hence, we typically have a non-linear minimization problem at hand.

Note that the basic structure of the problem is independent of the choice of a motion model. The model is in essence a statement about the function $\mathbf{u}(\mathbf{x})$. To make this explicit, we can write,

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}(\mathbf{x}; \mathbf{p}_m), \tag{2}$$

where $\mathbf{p}_m$ is a vector representing the model parameters.

A standard numerical approach for solving such a problem is to apply Newton's method. However, for errors which are sum of squares a good approximation to Newton's method is the Gauss-Newton method, which uses a first order expansion of the individual error quantities before squaring. If $\{\mathbf{u}\}_i$ current estimate of the flow field during the ith iteration, the incremental estimate $\{\delta\mathbf{u}\}$ can be obtained by minimizing the quadratic error measure

$$E(\{\delta\mathbf{u}\}) = \sum_{\mathbf{x}} (\Delta I + \nabla I \cdot \delta\mathbf{u}(\mathbf{x}))^2, \tag{3}$$

where

$$\Delta I(\mathbf{x}) = I(\mathbf{x},t) - I(\mathbf{x} - \mathbf{u}_i(\mathbf{x}), t-1),$$

that is the difference between the two images at corresponding pixels, after taking the current estimate into account.

As such, the minimization problem described in Equation 3 is underconstrained. The different motion models constrain the flow field in different ways. When these are used to describe the flow field, the estimation problem can be reformulated in terms of the unknown (incremental) model parameters. The details of these reformulations are described in the various sections corresponding to the individual motion models.

The third component, image warping, is achieved by using the current values of the model parameters to compute a flow field, and then using this flow field to warp $I(t-1)$ towards $I(t)$, which is used as the reference image. Our current warping algorithm uses bilinear interpolation. The warped image (*as against the original second image*) is then used for the computation of the error ΔI for further estimation[2]. The spatial gradient ∇I computations are based on the reference image.

The final component, coarse-to-fine refinement, propagates the current motion estimates from one level to the next level where they are then used as initial estimates. For the parametric component of the model, this is easy; the values of the parameters are simply transmitted to the next level. However, when a local model is also used, that information is typically in the form of a dense image (or images)—e.g., a flow field or a depth map. This image (or images) must be propagated via a pyramid expansion operation as described in [6]. The global parameters in combination with the local information can then be used to generate the flow field necessary to perform the initial warping at this next level.

3 Motion Models

3.1 Affine Flow

The Model: When the distance between the background surfaces and the camera is large, it is usually possible to approximate the motion of the surface as an affine transformation:

$$\begin{aligned} u(x,y) &= a_1 + a_2 x + a_3 y \\ v(x,y) &= a_4 + a_5 x + a_6 y \end{aligned} \tag{4}$$

Using vector notation this can be rewritten as follows:

$$\mathbf{u}(\mathbf{x}) = \mathbf{X}(\mathbf{x})\mathbf{a} \tag{5}$$

where $\mathbf{a}$ denotes the vector $(a_1, a_2, a_3, a_4, a_5, a_6)^T$, and

$$\mathbf{X}(\mathbf{x}) = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix}$$

Thus, the motion of the entire region is completely specified by the parameter vector $\mathbf{a}$, which is the unknown quantity that needs to be estimated.

The Estimation Algorithm: Let $\mathbf{a}_i$ denote the current estimate of the affine parameters. After using the flow field represented by these parameters in the warping step, an incremental estimate $\delta\mathbf{a}$ can be determined. To achieve this, we insert the parametric form of $\delta\mathbf{u}$ into Equation 3, and obtain an error measure that is a function of $\delta\mathbf{a}$.

$$E(\delta\mathbf{a}) = \sum_{\mathbf{x}} \left(\Delta I + (\nabla I)^T \mathbf{X}\delta\mathbf{a}\right)^2 \tag{6}$$

Minimizing this error with respect to $\delta\mathbf{a}$ leads to the equation:

$$\left[\sum \mathbf{X}^T(\nabla I)(\nabla I)^T\mathbf{X}\right]\delta\mathbf{a} = -\sum \mathbf{X}^T(\nabla I)(\Delta I). \tag{7}$$

[2] We have avoided using the standard notation I_t in order to avoid any confusion about this point.

Experiments with the affine motion model: To demonstrate use of the affine flow model, we show its performance on an aerial image sequence. A frame of the original sequence is shown in Figure 2a and the unprocessed difference between two frames of this sequence is shown in Figure 2b. Figure 2c shows the result of estimating an affine transformation using the hierarchical warp motion approach, and then using this to compensate for camera motion induced flow. Although the terrain is not perfectly flat, we still obtain encouraging compensation results. In this example the simple difference between the compensated and original image is sufficient to detect and locate a helicopter in the image. We use extensions of the approach, like integration of compensated difference images over time, to detect smaller objects moving more slowly with respect to the background.

3.2 Planar Surface Flow

The Model: It is generally known that the instantaneous motion of a planar surface undergoing rigid motion can be described as a second order function of image coordinates involving eight independent parameters (e.g., see [15]). In this section we provide a brief derivation of this description and make some observations concerning its estimation.

We begin by observing that the image motion induced by a rigidly moving object (in this case a plane), can be written as:

$$\mathbf{u}(\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\mathbf{A}(\mathbf{x})\mathbf{t} + \mathbf{B}(\mathbf{x})\omega \tag{8}$$

where $Z(\mathbf{x})$ is the distance from the camera of the point (i.e., depth) whose image position is $(\mathbf{x})$, and

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}$$

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} (xy)/f & -(f+x^2)/f & y \\ (f+y^2)/f & -(xy)/f & -x \end{bmatrix}.$$

The $\mathbf{A}$ and the $\mathbf{B}$ matrices depend only on the image positions and the focal length f and not on the unknowns: $\mathbf{t}$, the translation vector, ω the angular velocity vector, and Z.

A planar surface can be described by the equation

$$k_1 X + k_2 Y + k_3 Z = 1 \tag{9}$$

where (k_1, k_2, k_3) relate to the surface slant, tilt, and the distance of the plane from the origin of the chose coordinate system (in this case, the camera origin). Dividing throughout by Z, we get

$$\frac{1}{Z} = k_1\frac{x}{f} + k_2\frac{y}{f} + k_3.$$

Using $\mathbf{k}$ to denote the vector (k_1, k_2, k_3) and $\mathbf{r}$ to denote the vector $(x/f, y/f, 1)$ we obtain

$$\frac{1}{Z(\mathbf{x})} = \mathbf{r}(\mathbf{x})^T\mathbf{k}.$$

Substituting this into Equation 8 gives

$$\mathbf{u}(\mathbf{x}) = (\mathbf{A}(\mathbf{x})\mathbf{t})\,(\mathbf{r}(\mathbf{x})^T\mathbf{k}) + \mathbf{B}(\mathbf{x})\omega \tag{10}$$

This flow field is quadratic in $(\mathbf{x})$ and can be written also as

$$\begin{aligned} u(\mathbf{x}) &= a_1 + a_2 x + a_3 y + a_7 x^2 + a_8 xy \\ v(\mathbf{x}) &= a_4 + a_5 x + a_6 y + a_7 xy + a_8 y^2 \end{aligned} \tag{11}$$

where the 8 coefficients $(a_1, \ldots, a_8)$ are functions of the motion paramters $\mathbf{t}, \omega$ and the surface parmeters $\mathbf{k}$. Since this 8-parameter form is rather well-known (e.g., see [15]) we omit its details.

If the egomotion parameters are known, then the three parameter vector $\mathbf{k}$ can be used to represent the motion of the planar surface. Otherwise the 8-parameter representation can be used. In either case, the flow field is a linear in the unknown parameters.

The problem of estimating planar surface motion has been has been extensively studied before [21, 1, 23]. In particular, Negahdaripour and Horn [21] suggest iterative methods for estimating the motion and the surface parameters, as well as a method of estimating the 8 parameters and then decomposing them into the five rigid motion parameters the three surface parameters in closed form. Besides the embedding of these computations within the hierarchical estimation framework, we also take a slightly different, approach to the problem.

We assume that the rigid motion parameters are already known or can be estimated (e.g., see Section 3.3 below). Then, the problem reduces to that of estimating the three surface parameters $\mathbf{k}$. There are several practical reasons to prefer this approach: First, in many situations the rigid motion model may be more globally applicable than the planar surface model, and can be estimated using information from all the surfaces undergoing the same rigid motion. Second, unless the region of interest subtends a significant field of view, the second order components of the flow field will be small, and hence the estimation of the eight parameters will be inaccurate and the process may be unstable. On the other hand, the information concerning the three parameters $\mathbf{k}$ is contained in the first order components of the flow field, and (if the rigid motion parameters are known) their estimation will be more accurate and stable.

The Estimation Algorithm: Let $\mathbf{k}_i$ denote the current estimate of the surface parameters, and let $\mathbf{t}$ and ω denote the motion parameters. These parameters are used to construct an initial flow field that is used in the warping step. The residual information is then used to determine an incremental estimate $\delta\mathbf{k}$.

By substituting the parametric form of $\delta\mathbf{u}$

$$\begin{aligned} \delta\mathbf{u} &= \mathbf{u} - \mathbf{u_0} \\ &= (\mathbf{A}(\mathbf{x})\mathbf{t})\,(\mathbf{r}(\mathbf{x})^T(\mathbf{k_0} + \delta\mathbf{k})) + \mathbf{B}(\mathbf{x})\omega - (\mathbf{A}(\mathbf{x})\mathbf{t})\,(\mathbf{r}(\mathbf{x})^T\mathbf{k_0}) + \mathbf{B}(\mathbf{x})\omega \\ &= (\mathbf{A}(\mathbf{x})\mathbf{t})\,\mathbf{r}(\mathbf{x})^T\delta\mathbf{k} \end{aligned} \tag{12}$$

in Equation 3, we can obtain the incremental estimate $\delta\mathbf{k}$ as the vector that minimizes:

$$E(\delta\mathbf{k}) = \sum_{\mathbf{x}} \left((\Delta I + (\nabla I)^T(\mathbf{A}\mathbf{t})\mathbf{r}^T\delta\mathbf{k} \right)^2 \tag{13}$$

Minimizing this error leads to the equation:

$$\left[\sum \mathbf{r}(\mathbf{t}^T\mathbf{A}^T)(\nabla I)(\nabla I)^T(\mathbf{A}\mathbf{t})\mathbf{r}^T) \right] \delta\mathbf{k} = -\sum \mathbf{r}(\mathbf{t}^T\mathbf{A}^T)(\nabla I)\Delta I \tag{14}$$

This equation can be solved to obtain the incremental estimate $\delta\mathbf{k}$.

Experiments with the planar surface motion model: We demonstrate the application of the planar surface model using images from an outdoor sequence. One of the input images is shown in Figure 3a, and the difference between both input images is shown in Figure 3b. After estimating the camera motion between the images using the algorithm described in Section 3.3, we applied the planar surface estimation algorithm to a manually selected image window placed roughly over a region on the ground plane. These parameters were then used to warp the second frame towards the first (this process should align the ground plane alone). The difference between this warped image and the original image is shown in Figure 3c. The figure shows compensation of the ground plane motion, leaving residual parallax motion of the trees and other objects in the background. Finally, in order to demonstrate the plane-fit, we graphically projected a rectangular grid onto that plane. This is shown superimposed on the input image in Figure 3d.

3.3 Rigid Body Model

The Model: The motion of arbitrary surfaces undergoing rigid motion cannot usually be described by a single global model. We can however make use of the global rigid body model if we combine it with a local model of the surface. In this section, we provide a brief derivation of the global and the local models. Hanna [12] provides further details and results, and also describes how the local and global models interact at corner-like and edge-like image structures.

As described in Section 3.2, the image motion induced by a rigidity moving object can be written as:

$$\mathbf{u}(\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\mathbf{A}(\mathbf{x})\mathbf{t} + \mathbf{B}(\mathbf{x})\omega \tag{15}$$

where $Z(\mathbf{x})$ is the distance from the camera of the point (i.e., its depth), whose image position is $(\mathbf{x})$, and

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}$$

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} (xy)/f & -(f^2+x^2)/f & y \\ (f^2+y^2)/f & -(xy)/f & -x \end{bmatrix}.$$

The $\mathbf{A}$ and the $\mathbf{B}$ matrices depend only on the image positions and the focal length f and not on the unknowns: $\mathbf{t}$, the translation vector, ω the angular velocity vector, and Z. Equation 15 relates the parameters of the global model, ω and $\mathbf{t}$, with parameters of the local scene structure, $Z(\mathbf{x})$.

A local model we use is the frontal-planar model, which means that over a local image patch, we assume that $Z(\mathbf{x})$ is constant. An alternative model uses the assumption that $\delta Z(\mathbf{x})$—the difference between a previous estimate and a refined estimate—is constant over each local image patch.

We refine the local and global models in turn using initial estimates of the local structure parameters, $Z(\mathbf{x})$, and the global rigid body parameters ω and $\mathbf{t}$. This local/global refinement is iterated several times.

The Estimation Algorithm: Let the current estimates be denoted as $Z_i(\mathbf{x})$, $\mathbf{t}_i$ and ω_i. As in the other models, we can use the model parameters to construct an initial flow field, $\mathbf{u}_i(\mathbf{x})$, which is used to warp one of the image frames towards the next. The residual error between the warped image and the original image to which it is warped is used to

refine the parameters of the local and global models. We now show how these models are refined.

We begin by writing equation 15 in an incremental form so that

$$\delta\mathbf{u}(\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\mathbf{A}(\mathbf{x})\mathbf{t} + \mathbf{B}(\mathbf{x})\omega - \frac{1}{Z_0(\mathbf{x})}\mathbf{A}(\mathbf{x})\mathbf{t_0} - \mathbf{B}(\mathbf{x})\omega_0 \tag{16}$$

Inserting the parametric form of $\delta\mathbf{u}$ into Equation 3 we obtain the pixel-wise error as

$$E(\mathbf{t},\omega,1/Z(\mathbf{x})) = \left(\Delta I + (\nabla I)^T\mathbf{A}\mathbf{t}/Z(\mathbf{x}) + (\nabla I)^T\mathbf{B}\omega - (\nabla I)^T\mathbf{A}\mathbf{t}_i/Z_i(\mathbf{x}) - (\nabla I)^T\mathbf{B}\omega_i\right)^2. \tag{17}$$

To refine the local models, we assume that $1/Z(\mathbf{x})$ is constant over 5×5 image patches centered on each image pixel. We then algebraically solve for this Z both in order to estimate its current value, and to eliminate it from the global error measure. Consider the local component of the error measure,

$$E_{local} = \sum_{5\times5} E(\mathbf{t},\omega,1/Z(\mathbf{x})). \tag{18}$$

Differentiating equation 17 with respect to $1/Z(\mathbf{x})$ and setting the result to zero, we get

$$1/Z(\mathbf{x}) = \frac{-\sum_{5\times5}(\nabla I)^T\mathbf{A}\mathbf{t}\left(\Delta I - (\nabla I)^T\mathbf{A}\mathbf{t}_i/Z_i(\mathbf{x}) + (\nabla I)^T\mathbf{B}\omega - (\nabla I)^T\mathbf{B}\omega_i\right)}{\sum_{5\times5}\left((\nabla I)^T\mathbf{A}\mathbf{t}\right)^2} \tag{19}$$

To refine the global model, we minimize the error in Equation 17 summed over the entire image:

$$E_{global} = \sum_{Image} E(\mathbf{t},\omega,1/Z(\mathbf{x})). \tag{20}$$

We insert the expression for $1/Z(\mathbf{x})$ given in Equation 19—*not the current numerical value of the local parameter*—into Equation 20. The result is an expression for E_{global} that is non-quadratic in $\mathbf{t}$ but quadratic in ω . We recover refined estimates of $\mathbf{t}$ and ω by performing one Gauss-Newton minimization step using the previous estimates of the global parameters, $\mathbf{t_i}$ and $\omega_\mathbf{i}$, as starting values. Expressions are evaluated numerically at $t = t_i$ and $\omega = \omega_i$.

We then repeat the estimation algorithm several times at each image resolution.

Experiments with the rigid body motion model: We have chosen an outdoor scene to demonstrate the rigid body motion model. Figure 4a shows one of the input images, and Figure 4b shows the difference between the two input images. The algorithm was performed beginning at level 3 (subsampled by a factor of 8) of a Laplacian pyramid. The local surface parameters $1/Z(\mathbf{x})$ were all initialized to zero, and the rigid-body motion parameters were initialized to $\mathbf{t_0} = (0,0,1)^\top$ and $\omega = (0,0,0)^\top$. The model parameters were refined 10 times at each image resolution. Figure 4c shows the difference image between the second image and the first image after being warped using the final estimates of the rigid-body motion parameters and the local surface parameters. Figure 4d shows an image of the recovered local surface parameters $1/Z(\mathbf{x})$ such that bright points are nearer the camera than dark points. The recovered inverse ranges are plausible almost everywhere, except at the image border and near the recovered focus of expansion. The bright dot at the bottom right hand side of the inverse range map corresponds to a leaf in the original image that is blowing across the ground towards the camera. Figure 4e

shows a table of rigid-body motion parameters that were recovered at the end of each resolution of analysis.

More experimental results and a detailed discussion of the algorithm's performance on various types of scenes can be found in [12].

3.4 General Flow Fields

The Model: Unconstrained general flow fields are typically not described by any global parametric model. Different local models have been used to facilitate the estimation process, including constant flow within a local window and locally smooth or continuous flow. The former facilitates direct local estimation [18, 20], whereas the latter model requires iterative relaxation techniques [16] It is also not uncommon to use the combination of these two types of local models (e.g., [3, 10]).

The local model chosen here is constant flow within 5×5 pixel windows at each level of the pyramid. This is the same model as used by Lucas and Kanade [18] but here it is embedded as a local model within the hierarchical estimation framework.

The Estimation Algorithm: Assume that we have an approximate flow field from previous levels (or previous iterations at the same level). Assuming that the incremental flow vector $\delta\mathbf{u}$ is constant within the 5×5 window, Equation 3 can be written as

$$E(\delta u) = \sum_{\mathbf{x}} (\Delta I + \nabla I^T \delta\mathbf{u})^2 \tag{21}$$

where the sum is taken within the 5×5 window. Minimizing this error with respect to $\delta\mathbf{u}$ leads to the equation,

$$\left[\sum (\nabla I)(\nabla I)^T\right] \delta\mathbf{u} = -\sum \nabla I \Delta I. \tag{22}$$

We make some observations concerning the singularities of this relationship. If the summing window consists of a single element, the 2×2 matrix on the left-hand-side is an outer product of a 2×1 vector and hence has a rank of atmost unity. In our case, when the summing window consists of 25 points, the rank of the matrix on the left-hand-side will be two unless the directions of the gradient vectors ∇I everywhere within the window coincide. This situation is the general case of the *aperture effect*.

In our implementation of this technique, the flow estimate at each point is obtained by using a 5×5 windows centered around that point. This amounts to assuming implicitly that the flow field varies smoothly over the image.

Experiments with the general flow model: We demonstrate the general flow algorithm on an image sequence containing several independently moving objects, a case for which the other motion models described here are not applicable. Figure 5a shows one image of the original sequence. Figure 5b shows the difference between the two frames that were used to compute image flow. Figure 5c shows little difference between the compensated image and the other original image. Figure 5d shows the horizontal component of the computed flow field, and figure 5e shows the vertical component. In local image regions where image structure is well-defined, and where the local image motion is simple, the recovered motion estimates appear plausible. Errors predictably occur however at motion boundaries. Errors also occur in image regions where the local image structure is not well-defined (like some parts of the road), but for the same reason, such errors do not appear as intensity errors in the compensated difference image.

4 Discussion

Thus far, we have described a hierarchical framework for the estimation of image motion between two images using various models. Our motivation was to generalize the notion of direct estimation to model-based estimation and unify a diverse set of model-based estimation algorithms into a single framework. The framework also supports the combined use of parametric global models and local models which typically represent some type of a smoothness or local uniformity assumption.

One of the unifying aspects of the framework is that the same objective function (SSD) is used for all models, but the minimization is performed with respect to different parameters. As noted in the introduction, this is enabled by viewing all these problems from the perspective of image registration.

It is interesting to contrast this perspective (of model-based image registration) with some of the more traditional approaches to motion analysis. One such approach is to compute image flow fields, which involves combining the local brightness constraint with some sort of a global smoothness assumption, and then interpret them using appropriate motion models. In contrast, the approach taken here is to use the motion models to constrain the flow field computation. The obvious benefit of this is that the resulting flow fields may generally be expected to be more consistent with models than general smooth flow fields. Note, however, that the framework also includes general smooth flow field techniques, which can be used if the motion model is unknown.

In the case of models that are not fully parametric, local image information is used to determine local image/scene properties (e.g., the local range value). However, the accuracy of these can only be as good as the available local image information. For example, in homogeneous areas of the scene, it may be possible to achieve perfect registration even if the surface range estimates (and the corresponding local flow vectors) are incorrect. However, in the presence of significant image structures, these local estimates may be expected to be accurate. On the other hand, the accuracy of the global parameters (e.g., the rigid motion parameters) depends only on having sufficient and sufficiently diverse local information across the entire region. Hence, it may be possible to obtain reliable estimates of these global parameters, even though estimated local information may not be reliable everywhere within the region. For fully parametric models, this problem does not exist.

The image registration problem addressed in this paper occurs in a wide range of image processing applications, far beyond the usual ones considered in computer vision (e.g., navigation and image understanding). These include image compression via motion compensated encoding, spatiotemporal analysis of remote sensing type of images, image database indexing and retrieval, and possibly object recognition. One way to state this general problem is as that of recovering the coordinate system that relate two images of a scene taken from two different viewpoints. In this sense, the framework proposed here unifies motion analysis across these different applications as well.

Acknowledgements: Many individuals have contributed to the ideas and results presented here. These include Peter Burt and Leonid Oliker from the David Sarnoff Research Center, and Shmuel Peleg from Hebrew University.

References

1. G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(4):384–401, July 1985.
2. P. Anandan. A unified perspective on computational techniques for the measurement of visual motion. In *International Conference on Computer Vision*, pages 219–230, London, May 1987.
3. P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.
4. J. R. Bergen and E. H. Adelson. Hierarchical, computationally efficient motion estimation algorithm. *J. Opt. Soc. Am. A.*, 4:35, 1987.
5. J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg. Computing two motions from three frames. In *International Conference on Computer Vision*, Osaka, Japan, December 1990.
6. P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communication*, 31:532–540, 1983.
7. P.J. Burt, J.R. Bergen, R. Hingorani, R. Kolczinski, W.A. Lee, A. Leung, J. Lubin, and H. Shvaytser. Object tracking with a moving camera, an application of dynamic motion analysis. In *IEEE Workshop on Visual Motion*, pages 2–12, Irvine, CA, March 1989.
8. P.J. Burt, R. Hingorani, and R. J. Kolczynski. Mechanisms for isolating component patterns in the sequential analysis of multiple motion. In *IEEE Workshop on Visual Motion*, pages 187–193, Princeton, NJ, October 1991.
9. Stefan Carlsson. Object detection using model based prediction and motion parallax. In *Stockholm workshop on computational vision*, Stockholm, Sweden, August 1989.
10. J. Dengler. Local motion estimation with the dynamic pyramid. In *Pyramidal systems for computer vision*, pages 289–298, Maratea, Italy, May 1986.
11. W. Enkelmann. Investigations of multigrid algorithms for estimation of optical flow fields in image sequences. *Computer Vision, Graphics, and Image Processing*, 4339:150–177, 1988.
12. K. J. Hanna. Direct multi-resolution estimation of ego-motion and structure from motion. In *Workshop on Visual Motion*, pages 156–162, Princeton, NJ, October 1991.
13. J. Heel. Direct estimation of structure and motion from multiple frames. Technical Report 1190, MIT AI LAB, Cambridge, MA, 1990.
14. E. C. Hildreth. *The Measurement of Visual Motion.* The MIT Press, 1983.
15. B. K. P. Horn. *Robot Vision.* MIT Press, Cambridge, MA, 1986.
16. B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
17. B. K. P. Horn and E. J. Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, 2(1):51–76, June 1988.
18. B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130, 1981.
19. L. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image-sequences. In *International Conference on Computer Vision*, pages 199–213, Tampa, FL, 1988.
20. H. H. Nagel. Displacement vectors derived from second order intensity variations in intensity sequences. *Computer Vision, Pattern recognition and Image Processing*, 21:85–117, 1983.
21. S. Negahdaripour and B.K.P. Horn. Direct passive navigation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(1):168–176, January 1987.
22. A. Singh. An estimation theoretic framework for image-flow computation. In *International Conference on Computer Vision*, Osaka, Japan, November 1990.
23. A.M. Waxman and K. Wohn. Contour evolution, neighborhood deformation and global image flow: Planar surfaces in motion. *International Journal of Robotics Research*, 4(3):95–108, Fall 1985.

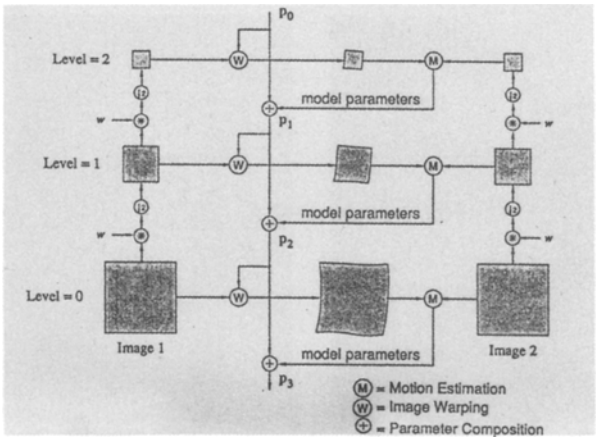


Fig. 1. Diagram of the hierarchical motion estimation framework.

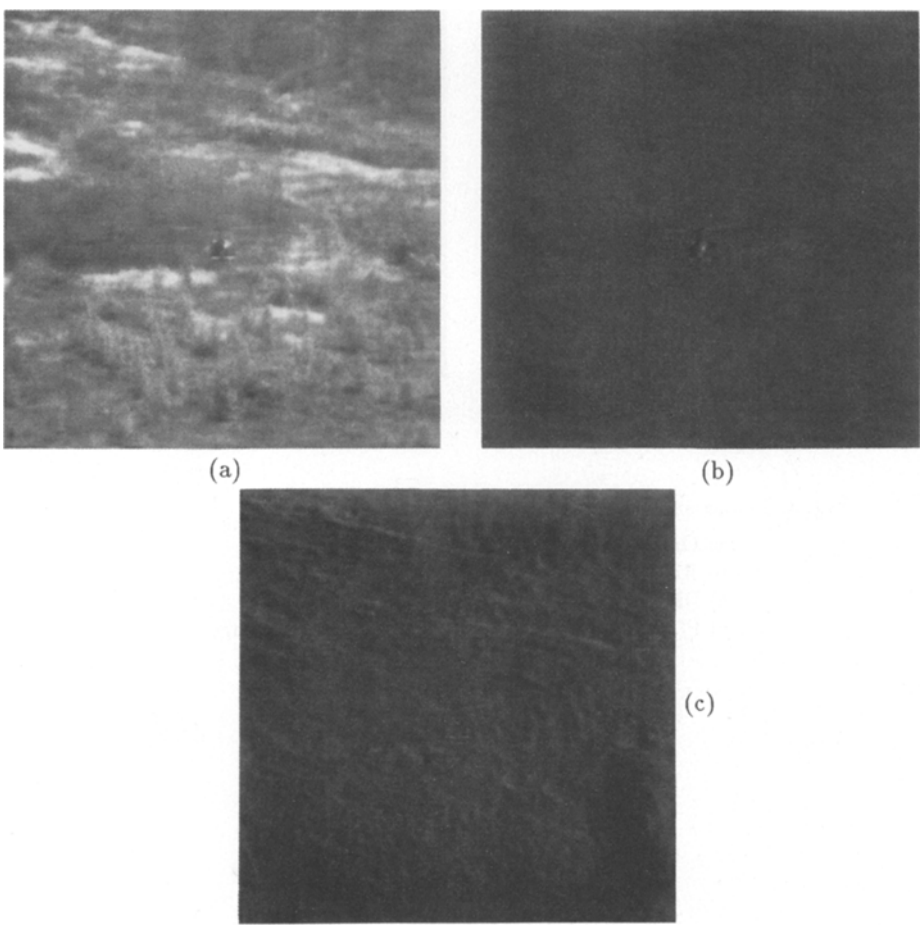


Fig. 2. Affine motion estimation: a) Original. b) Raw difference. c) Compensated difference.

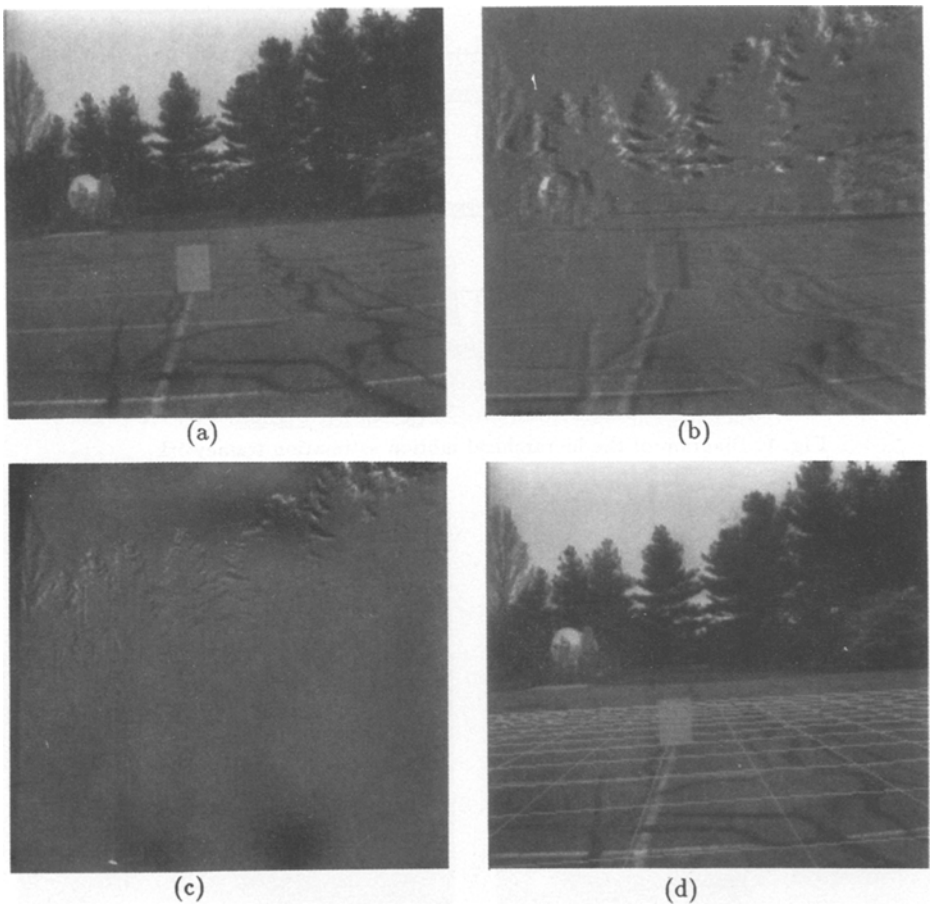
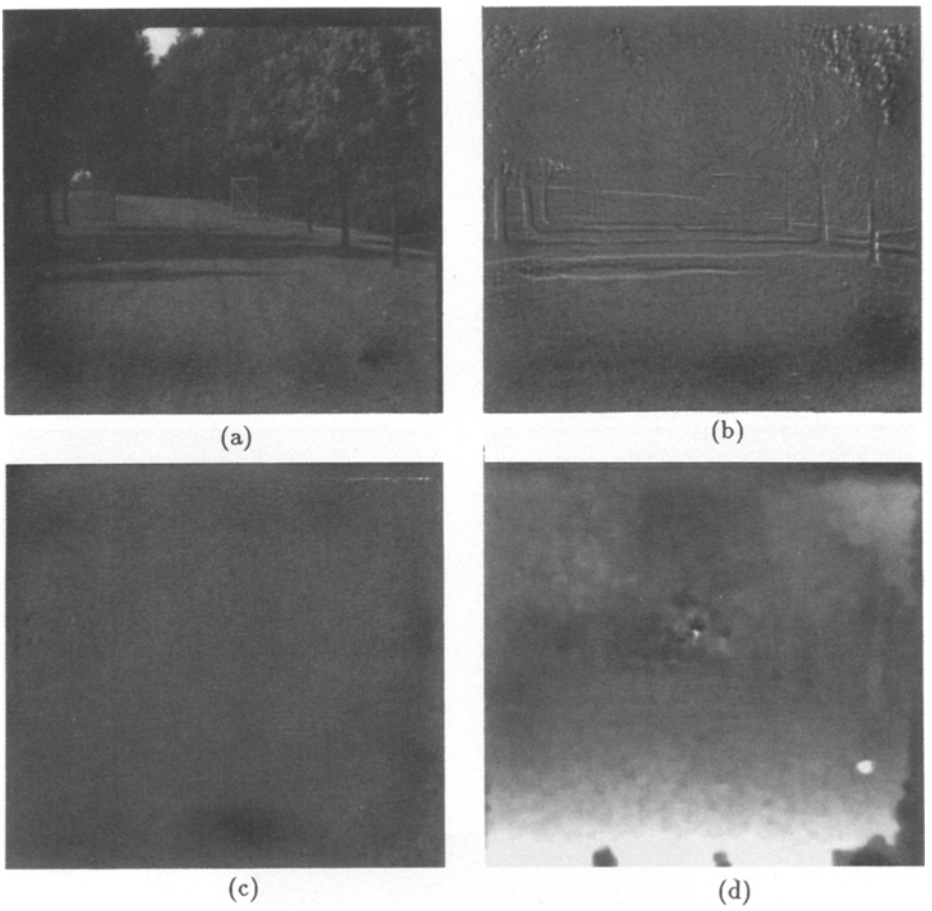


Fig. 3. Planar surface motion estimation.

a) Original image.
b) Raw difference.
c) Difference after planar compensation.
c) Planar grid superimposed on the original image.



Resolution	Ω	T
.	(.0000,.0000,.0000)	(.0000,.0000,1.0000)
32 × 30	(.0027,.0039,-.0001)	(-.3379,-.1352,.9314)
64 × 60	(.0038,.0041,.0019)	(-.3319,-.0561,.9416)
128 × 120	(.0037,.0012,.0008)	(-.0660,-.0383,.9971)
256 × 240	(.0029,.0006,.0013)	(-.0255,-.0899,.9956)

Fig. 4. Egomotion based flow model.

a) Original image from an outdoor sequence.
b) Raw difference.
c) Difference after ego-motion compensation.
d) Inverse range map.
e) Rigid body parameters recovered at each resolution.

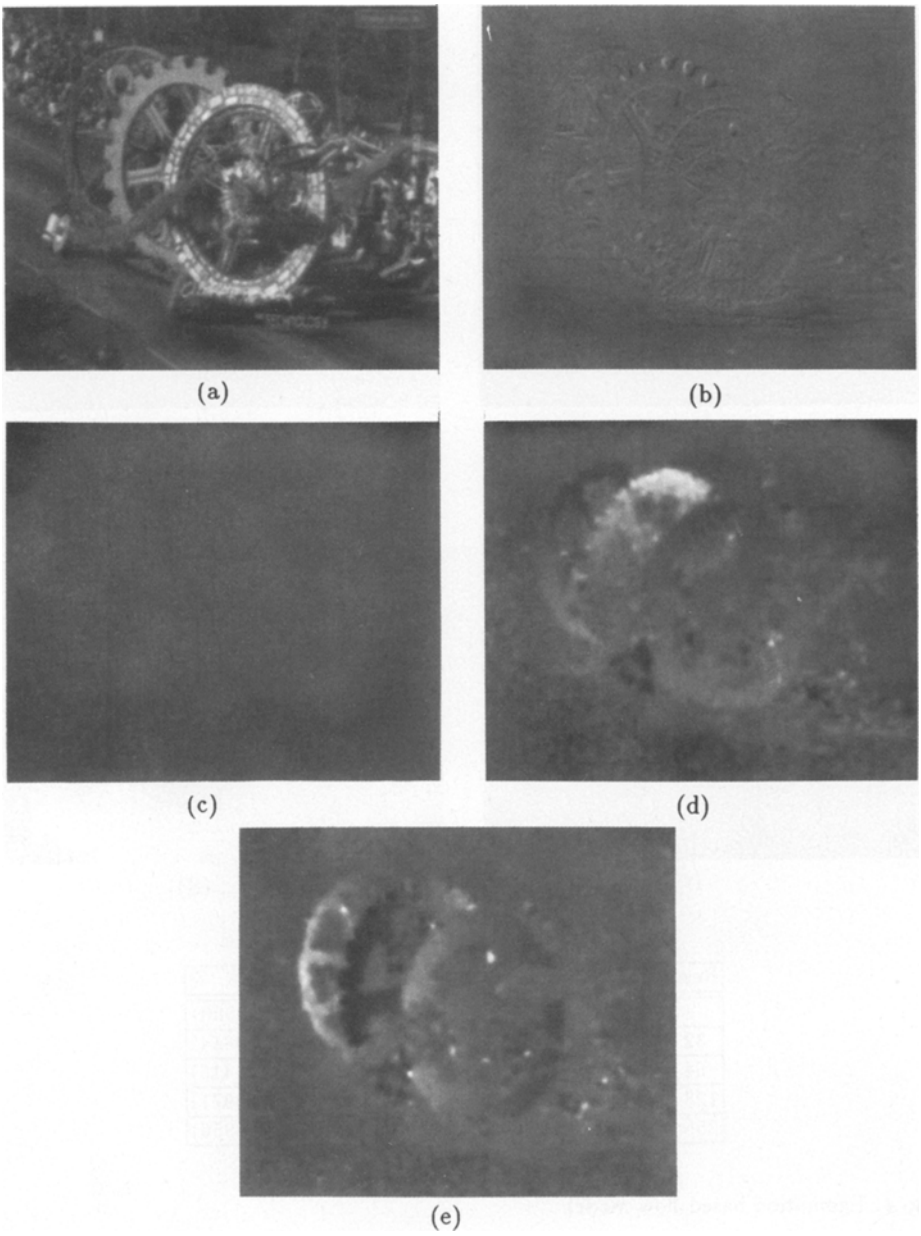


Fig. 5. Optical flow estimation.

a) Original image.
b) Raw difference.
c) Difference after motion compensation.
d) Horizontal component of the recovered flow field.
e) Vertical component of the recovered flow field.