Design and Use of Linear Models for Image Motion Analysis

David J. Fleet[*][†] Michael J. Black[†] Yaser Yacoob[‡] Allan D. Jepson[§]

[*] Department of Computing and Information Science, Queen's University, Kingston, Ontario K7L 3N6

[†] Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304

[‡] Computer Vision Laboratory, University of Maryland, College Park, MD 20742

[§] Department of Computer Science, University of Toronto, Toronto, Ontario M5S 1A4

fleet@cs.queensu.ca, black@parc.xerox.com, yaser@cs.umd.edu,
jepson@vis.toronto.edu

Abstract

Linear parameterized models of optical flow, particularly affine models, have become widespread in image motion analysis. The linear model coefficients are straightforward to estimate, and they provide reliable estimates of the optical flow of smooth surfaces. Here we explore the use of parameterized motion models that represent much more varied and complex motions. Our goals are threefold: to construct linear bases for complex motion phenomena; to estimate the coefficients of these linear models; and to recognize or classify image motions from the estimated coefficients. We consider two broad classes of motions: i) generic "motion features" such as motion discontinuities and moving bars; and ii) non-rigid, object-specific, motions such as the motion of human mouths. For motion features we construct a basis of *steerable flow fields* that approximate the motion features. For object-specific motions we construct basis flow fields from example motions using principal component analysis. In both cases, the model coefficients can be estimated directly from spatiotemporal image derivatives with a robust, multi-resolution scheme. Finally, we show how these model coefficients can be use to detect and recognize specific motions such as occlusion boundaries and facial expressions.

Index terms: Optical flow, motion discontinuities, occlusion, steerable filters, learning, eigenspace methods, motion-based recognition, non-rigid and articulated motion.

Paper Title:

Design and Use of Linear Models for Image Motion Analysis

Authors:

D. Fleet, M. Black, Y. Yacoob, A. Jepson

Index Terms:

Optical flow, motion discontinuities, occlusion, steerable filters, learning, eigenspace methods, motion-based recognition, non-rigid and articulated motion.

Contact Information:

David J. Fleet
Xerox Palo Alto Research Center
3333 Coyote Hill Rd
Palo Alto, CA 94304

Phone: (650) 812–4267
FAX: (650) 812–4334
Email: fleet@parc.xerox.com

1 Introduction

Linear parameterized models of optical flow play a significant role in motion *estimation* and motion *explanation*. They facilitate estimation by enforcing strong constraints on the spatial variation of the image motion within a region. Because they pool hundreds or thousands of motion constraints to estimate a much smaller number of model parameters, they generally provide accurate and stable estimates of optical flow. Moreover, the small number of parameters provide a concise description of the image motion which is useful for explanation; for example, parameterized models of optical flow have been used to recognize facial expressions from image sequences [15].

Translational and affine models have been used successfully for estimating and representing the optical flow of smooth textured surfaces [7, 21, 26, 27, 39, 40, 59]. These models have been applied locally within small image regions, and globally, for applications such as image stabilization and mosaicing. Low-order polynomial models have a common mathematical form, where the optical flow field, $\mathbf{u}(\mathbf{x};\, \mathbf{c})$, over positions $\mathbf{x} = (x,\, y)$ can be written as a weighted sum of *basis flow fields*:

$$\mathbf{u}(\mathbf{x};\, \mathbf{c}) \;=\; \sum_{j=1}^{n} c_j \, \mathbf{b}_j(\mathbf{x}) \;, \tag{1}$$

where $\{\mathbf{b}_j(\mathbf{x})\}_{j=1...n}$ is the basis set and $\mathbf{c} = (c_1, ..., c_n)$ is the vector containing the scalar coefficients. A translational model requires two basis flow fields, encoding horizontal and vertical translation, while affine models require six basis flow fields, as shown in Figure 1. With this linear form (1), the model coefficients can be estimated directly from the spatiotemporal derivatives of image intensity in a stable, efficient, manner. In particular, the gradient constraint equation, derived by linearizing the brightness constancy constraint, is linear in the motion coefficients [7, 40].

But the use of such models is limited to motions for which the models are good approximations to the actual optical flow. Affine models account for the motion of a planar surface under orthographic projection and provide a reasonable approximation to the motions of smooth surfaces in small image regions. But they have limited applicability to complex natural scenes. For example, many image regions contain multiple image motions because of moving occlusion boundaries, transparency, reflections, or independently moving objects. Many natural scenes also contain complex local patterns of optical flow.

A great deal of work has been devoted to extending parameterized models to cope with multiple motions [1, 2, 8, 12, 24, 37, 38, 58]. By varying the spatial support of the model according to the expected smoothness of the flow [53], using robust statistical techniques [2, 45, 12] or mixture models [1, 37, 38, 55, 61, 60], or by employing layered representations [58], researchers have

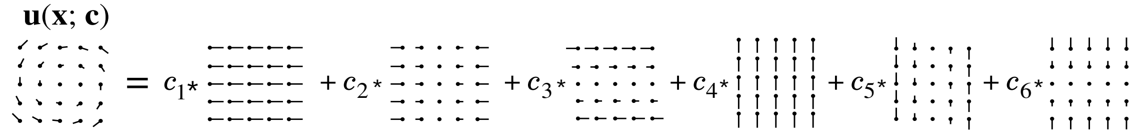


Figure 1: Affine motion model expressed as a linear sum of orthogonal basis flows. As with flow fields show below, the black dot denotes the origin of the flow vector, and the length and direction of the line segment reflect the speed and the direction of image velocity.

been able to apply simple parameterized models to a reasonably wide variety of situations. Some researchers have extended regression-based flow techniques beyond low-order polynomial models to overlapping splines [52] and wavelets [62], but most have concentrated on the estimation of optical flow fields that arise from the motion of smooth surface patches.

Complex motions, like those in Figure 2, remain problematic. But, while each type of motion in Figure 2 is more complex than affine, each is also highly constrained. For instance, mouths are physically constrained, performing a limited class of motions, yet they pose a challenge for optical flow techniques. Having a model of the expected motion of mouths would both improve flow estimation and provide a rich description of the motion that might aid subsequent interpretation.

This paper concerns how one can explicitly model certain classes of complex motions, like those in Figure 2(a,b), using linear parameterized models. We address three main problems, namely, model construction, optical flow estimation, and the detection of model occurrences. A key insight is that many complex motions can be modeled and estimated in the same way as a conventional affine model; that is, as linear parameterized models. The construction of a parameterized model amounts to finding an appropriate basis $\{\mathbf{b}_j(\mathbf{x})\}_{j=1\ldots n}$. Here, we describe how to construct models for *generic* types of *motion features*, such as moving edges (occlusion boundaries) and moving bars, and for *domain-specific* motions like the motion of human mouths. For some motion classes, e.g. discontinuities, we compute optimal linear basis sets directly from a generative model. In others, e.g. human mouth motion, explicit generative models may not be available, and we show how to construct approximate models from example motions using principal component analysis.

The resulting models can be used for optical flow estimation, as well as motion-based recognition and tracking. To compute optical flow with linear parameterized models we *directly* estimate the model coefficients from spatiotemporal image derivatives using a conventional area-based regression technique. We use a robust, coarse-to-fine, gradient-based algorithm. If the fitted model provides a good description of the time-varying image, then one can also use the estimated coefficients of the model for detection and interpretation of instances of a particular motion class. Here we provide examples of the detection of motion edges, moving bars, as well as the recognition of

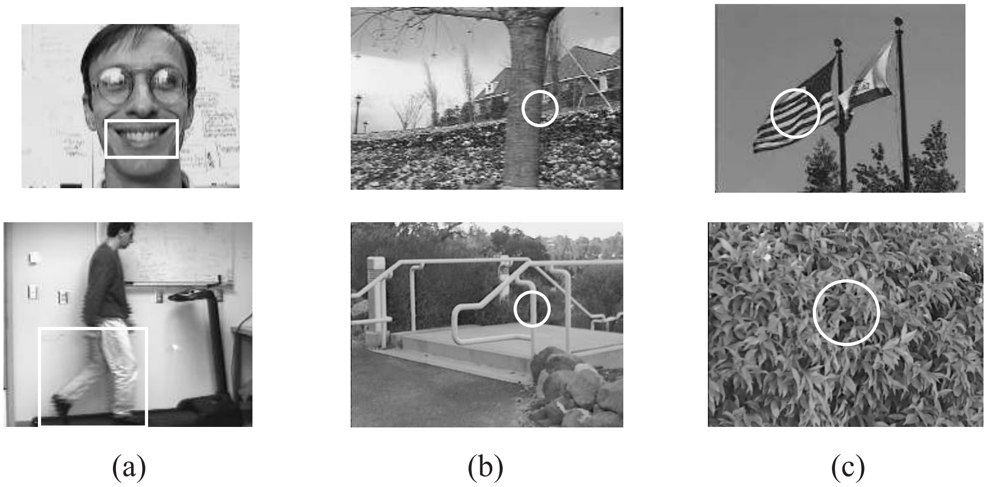


Figure 2: Examples of scenes with complex motions (delineated by the white borders). (a) Non-rigid and articulated human motion. (b) Generic motion features include motion discontinuities and moving bars. (c) “Textural” motion of cloth and plants.

simple speech and walking motions.

2 Problem Context and Related Work

Below we review previous work that addresses complex motions such as those depicted in Figure 2, namely, *generic motion features*, *domain-specific motion models*, and *motion texture*.

Generic Motion Features: Early approaches to occlusion detection involved the estimation of dense optical flow, from which motion discontinuities were then detected. Authors explored region or edge segmentation techniques [47, 54] and analyzed the distribution of flow vectors in local neighborhoods [51]. These methods are often unreliable because they require accurate estimates of optical flow near the motion boundary, where optical flow is difficult to estimate.

Other approaches have explored the local spatiotemporal image structure at a motion discontinuity [22, 28, 44]. For example, Fleet and Langley [28] examined the structure of occlusions and motion discontinuities in the frequency domain (also see [5]) and discussed detection methods based on second-order motion processing. Black and Anandan [11] looked for multiple peaks in a sum-of-squared difference surface as evidence of an occlusion/disocclusion boundary. These methods do not explicitly model the image motion present at a motion feature, and have not proved sufficiently reliable in practice.

More recent optical flow techniques estimate piecewise smooth flow fields using line processes or robust statistics that treat motion discontinuities as violations of a spatial smoothness assumption

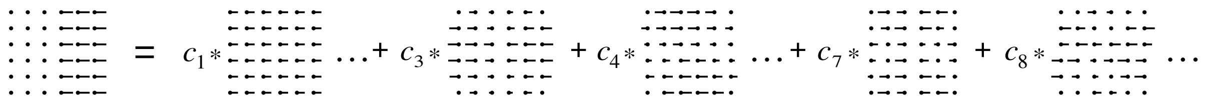


Figure 3: A motion discontinuity can be approximated by a weighted sum of basis flow fields.

[12, 34, 35, 50]. Although these methods work with motion discontinuities in many instances, they embody a relatively weak model of discontinuities and they do not generalize easily to more complex features such as moving bars.

In contrast to previous approaches, here we construct explicit (approximate) models of motion features using linear combinations of basis flow fields as illustrated in Figure 3. Estimating the image motion with these models is analogous to motion estimation with conventional affine models. In this way the coefficients of the model are recovered directly from the image data without first computing a dense optical flow field.

The proposed framework can be used for any motion that can be well approximated using a linear parameterized model. In this paper we experiment with moving edges and bars, like those in Figure 2(b). We refer to these as *motion features* to emphasize their relationship to static image features. Throughout the history of computer vision, image features such as lines, edges, and junctions have been treated as primitive structures in images upon which later stages of processing, such as segmentation and object recognition, are based. Although work on feature detection has focused on static images, such as edges and lines, there are analogous features in image sequences and binocular image pairs. Compared to static image features, these motion and stereo features typically convey direct information about scene structure. It is therefore important that we treat motion features as a source of structure to model, rather than a source of error in the estimation process.

Domain-Specific Motions: Figure 2(a) shows two other domains in which motion models would be useful both to constrain optical flow estimation and to facilitate interpretation and recognition. Motion estimation of mouths during speech is a challenge for conventional optical flow techniques because of the large image velocities that occur when the mouth is opening or closing. In addition, the appearance and disappearance of the tongue and mouth cavity are particularly problematic for techniques that assume intensity conservation [18]. For these reasons a model of mouth motion will help constrain the estimation of optical flow. The time-varying coefficients of the model may also provide an effective description of the flow that can be used for recognition.

Black and Yacoob [15] modeled the motion of a human face and facial features using parameterized flow models (planar, affine, and affine+curvature). They showed how simple models could represent a rich variety of image motions, and how the motion parameters could be used to recognize facial expressions. Another recent example of domain-specific motion models is the kinematic-based model of articulated human figures used to track people [19, 63]. In these approaches, linear parameterized models were designed so that the kinematic parameters could be extracted using a gradient-based motion constraint equation. But in both of these studies the motion models were hand-coded. In the case of human mouths, the hand-coded model proved too crude for the recognition of speech as it did not capture the natural variability of human mouths. In this paper, by comparison, we show how models of facial feature motion can be constructed from examples and used for the recognition of speech in highly constrained situations.

Much of the recent work on learning domain-specific models of image deformation has occurred in the face recognition literature, where the goal is to model the deformations between the faces of different people, or faces of a single person in different poses [9, 25, 33, 41, 56, 57]. Correspondences between different faces were obtained either by hand or by an optical flow method, and were then used to learn a low-dimensional model. In some cases this involved learning the parameters of a physically-based deformable object [41]. In others, a basis set of deformation vectors was obtained (e.g., see work by Hallinan [33] on learning "EigenWarps"). One of the main uses of the learned models has been view-based synthesis, as in [56, 57, 23] for example.

Related work has focused on learning the deformation of curves or parameterized curve models [4, 49]. Sclaroff and Pentland [49] estimated modes of deformation for silhouettes of non-rigid objects. Sclaroff and Isidoro [48] use a similar approach to model the deformation of image regions. Like our method they estimate the linear coefficients of the model directly from the image. Unlike our approach, they did not learn the basis flows from optical flow examples of the object being tracked, nor use the coefficients for detection or recognition.

Motion Texture: The flags and bush shown in Figure 2(c) illustrate another form of domain-specific motion, often called motion texture. Authors have explored texture models for synthesizing such textural motions and frequency analysis techniques for recognizing them [43]. These motions typically exhibit complex forms of local occlusion and self shadowing that violate the assumption of brightness constancy. They also tend to exhibit statistical regularities at specific spatial and temporal scales that make it difficult to find good linear approximations. Experiments using linear models to account for these motions [16] suggest that such models may not be appropriate, and they therefore remain outside the scope of the current work.

3 Constructing Parameterized Motion Models

The construction of a linear parameterized model for a particular motion class involves finding a set of *basis flow fields* that can be combined linearly to approximate flow fields in the motion class. In the case of motion features, such as motion edges and bars, we begin with the design of an idealized, generative model. From this model we explicitly construct an approximate basis set using steerable filters [29, 46], yielding a basis of *steerable flow fields*.

With complex object motions, such as mouths or bushes, no analytical model exists. If an ensemble of training flow fields is available, then one can use principal component analysis (PCA) or independent component analysis (ICA) [6] to find a set of basis flow fields. A similar approach was taken by Nayer *et al.* [42] to model edges, bars, and corners in static images.

3.1 Models for Motion Features Using Steerable Flow Fields

First, consider the modeling of motion edges and bars like those in Figure 4. The motion edge can be described by a mean (DC) motion vector $\mathbf{u}_t$, an edge orientation θ, and a velocity change across the motion boundary $\Delta\mathbf{u}$. Let $\mathbf{f}(\mathbf{x};\, \mathbf{u}_t, \Delta\mathbf{u}, \theta)$ be the corresponding flow field over spatial positions $\mathbf{x} = (x, y)$ in a circular image window R. Because $\mathbf{f}(\mathbf{x};\, \mathbf{u}_t, \Delta\mathbf{u}, \theta)$ is non-linear in the feature parameters, $\mathbf{u}_t$, $\Delta\mathbf{u}$, and θ, direct parameter estimation in the 5-dimensional space, without a good initial guess, can be difficult.

As depicted in Figure 3, our approach is to approximate $\mathbf{f}(\mathbf{x};\, \mathbf{u}_t, \Delta\mathbf{u}, \theta)$ by its projection onto a subspace spanned by a collection of n basis flow fields $\mathbf{b}_j(\mathbf{x})$,

$$\mathbf{f}(\mathbf{x};\, \mathbf{u}_t, \Delta\mathbf{u}, \theta) \;\approx\; \mathbf{u}(\mathbf{x};\, \mathbf{c}) \;=\; \sum_{j=1}^{n} c_j\, \mathbf{b}_j(\mathbf{x})\,. \tag{2}$$

Although the basis flow fields could be learned from examples using Principal Component Analysis, here we construct *steerable* sets of basis flow fields. These are similar to those learned using PCA up to rotations of invariant subspaces. The basis flow fields are steerable in orientation and velocity, and provide reasonably accurate approximations to the motion features of interest, namely, motion edges and motion bars. We first construct a basis for the spatial structure of the features. These bases are then combined (as a tensor product) with a basis for velocity to produce the basis flow fields. This approach also provides a straightforward mathematical relationship between the feature parameters and the model coefficients. This facilitates subsequent detection and parameter estimation.

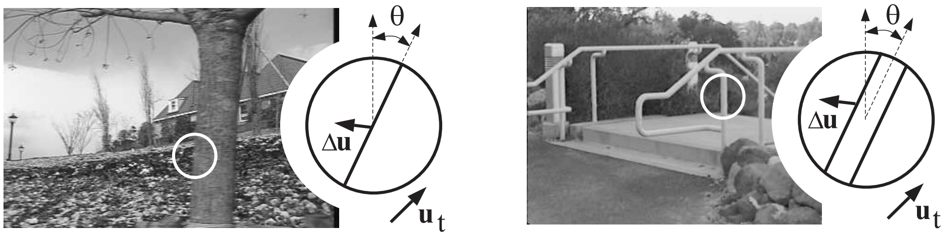


Figure 4: Example motion features and models for a motion discontinuity and a moving bar. The parameters of the idealized models are the mean (DC) translational velocity $\mathbf{u}_t$, the feature orientation θ, and the velocity change across the motion feature $\Delta\mathbf{u}$.

3.1.1 Spatial Bases for Edges and Bars

A generic template for the spatial structure of a motion boundary, $S(\mathbf{x})$, is given by the step edge in Figure 5 (left). This is a mean-zero, unit amplitude step edge within a circular, 32 pixel diameter, window. A circular window is used to avoid orientation anisotropies in the basis set.

A steerable basis that approximates $S(\mathbf{x})$ under 2D rotations is constructed using the method described in [46]. This yields a set of complex-valued basis functions, $b_k(\mathbf{x})$, at specific angular harmonics with wavenumber k. The real and imaginary parts of $b_k(\mathbf{x})$ form a quadrature pair, and for convenience, we normalize the basis images so $||b_k(\mathbf{x})|| = 1$. The features are then steered (rotated) by taking a linear combination of the basis functions with sinusoidal weights (steering functions). Thus, the edge template, $S(\mathbf{x})$, and rotated versions of it, $S_\theta(\mathbf{x})$, are approximated by a linear combination of the basis images,

$$S_\theta(\mathbf{x}) \approx \Re\left[\sum_{k\in K} \sigma_k \, a_k(\theta) \, b_k(\mathbf{x})\right] , \tag{3}$$

where $\theta \in [0, 2\pi)$ is the rotation angle, K is the set of angular wavenumbers used in the approximation, $a_k(\theta)$ are the steering functions, σ_k are real-valued weights on the different harmonics, and $\Re[z]$ denotes the real-part of z. The weights, σ_k, encode the relative magnitudes of the harmonics that best approximate the spatial structure of the edge. Because the basis images, $b_k(\mathbf{x})$, are unitary (orthogonal, unit norm), the weights σ_k are equal to the inner product of $S(\mathbf{x})$ and $b_k(\mathbf{x})$. To obtain the best approximation, K contains those wavenumbers with the largest weights [46]. Each basis function generated using this method is equal to the product of an isotropic radial function and a complex-valued angular sinusoid. As a consequence, the steering functions are angular harmonics,

$$a_k(\theta) = e^{-ik\theta} , \tag{4}$$

where θ is the rotation angle. Finally, we are only interested in the real part of the expression in (3) because the templates are real-valued; thus, one could also rewrite (3) in terms of real-valued

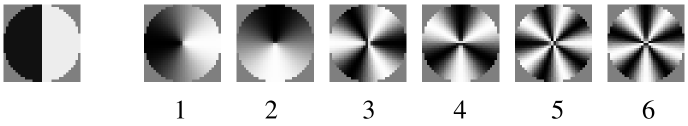


Figure 5: (left) Edge template $S(\mathbf{x})$. (right) Real and imaginary parts of the first three basis images.

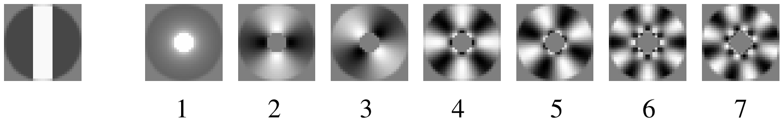


Figure 6: (left) Bar template $S(\mathbf{x})$. (right) Real and imaginary parts of the first four complex-valued basis images. The first has wavenumber zero; its imaginary part is zero and not shown. In this case the bar has a fixed width that is one quarter the diameter of the circular window. We do not consider bars at multiple scales in this paper.

bases and weights,

$$S_\theta(\mathbf{x}) \;\approx\; \sum_{k \in K} \sigma_k \left(\cos(k\theta) \Re[b_k(\mathbf{x})] \;+\; \sin(k\theta) \Im[b_k(\mathbf{x})] \right) \;,$$

where $\Re[b_k(\mathbf{x})]$ and $\Im[b_k(\mathbf{x})]$ are the real and imaginary parts of $b_k(\mathbf{x})$.

The basis set for the static edge structure includes a uniform intensity (DC) image and a set of images at nonzero angular wavenumbers. Because the edge is odd-symmetric, it contains only odd-numbered harmonics. The real and imaginary parts of the first three odd-numbered harmonics, ignoring the DC component, are shown in Figures 5(1–6). By comparison, the template for the spatial structure of a bar is shown in Figure 6. The template is mean-zero, and the bar has an amplitude of 1. The bar is 8 pixels wide, and the diameter of the circular window R is 32 pixels. The basis set for the bar is composed of even-numbered harmonics, the first four of which are shown in Figure 6 (1–7).

The quality of the approximation provided by the basis (3) is easily characterized by the fraction of the energy in $S(\mathbf{x})$ that is captured with the selected wavenumbers. The energy in the approximation is given by the sum of squared weights, σ_k^2, for wavenumbers in the approximation. If we let $K(n)$ be the set of wavenumbers that correspond to the n largest values of σ_k^2, then the quality of the approximation, as a function of n, is given by

$$Q(n) = \left(\sum_{k \in K(n)} \sigma_k^2 \right) \,/\, \left(\sum_{\mathbf{x} \in R} S(\mathbf{x})^2 \right) \;. \tag{5}$$

This quantity is shown in Figure 7 for the edge and the bar. In the case of the edge, the three harmonics shown in Figure 5 account for approximately 94% of the energy in the edge template. With the bar, the four harmonics shown in Figure 6 account for over 90% of the energy.

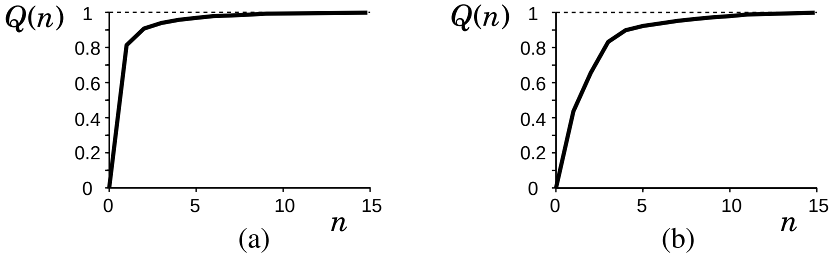


Figure 7: Fraction of energy in the edge model (a) and the bar model (b) that is captured in the linear approximation as a function of the number of complex-valued basis functions used.

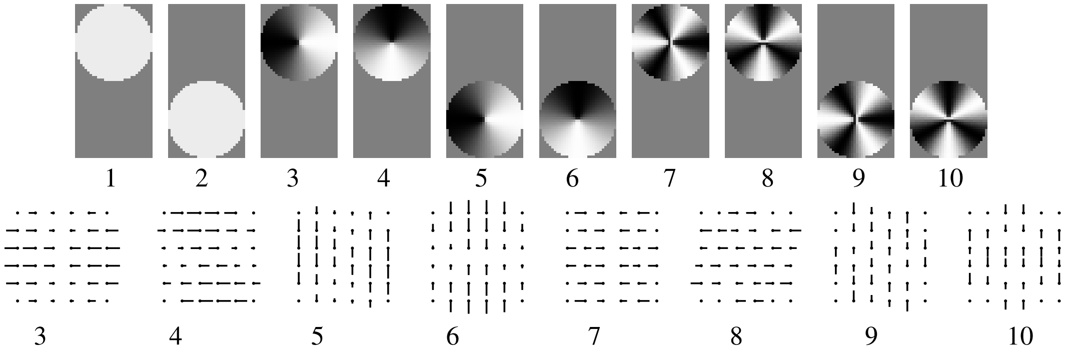


Figure 8: Steerable basis flow fields for occluding edges. (top) Flow fields are depicted as images in which the horizontal (u) component of the flow is the top half, and the vertical (v) component is the bottom half. Black indicates motion to the left or up respectively. Gray is zero motion and white is motion to the right or down. (bottom) Basis flow fields (3–10) depicted as subsampled vector fields.

3.1.2 Basis Flow Fields for Motion Edges and Bars

The basis flow fields for the motion features are formed by combining a basis for velocity with the basis for the spatial structure of the features. Two vectors, $(1, 0)^T$ and $(0, 1)^T$, provide a basis for translational flow. The basis flow fields for the horizontal and vertical components of the motion features are therefore given by

$$\mathbf{b}_k^h(\mathbf{x}) = \begin{pmatrix} b_k(\mathbf{x}) \\ 0 \end{pmatrix} \ , \quad \mathbf{b}_k^v(\mathbf{x}) = \begin{pmatrix} 0 \\ b_k(\mathbf{x}) \end{pmatrix} \ . \tag{6}$$

For each angular wavenumber, k, there are four real-valued flow fields. These are the real and imaginary parts of $b_k(\mathbf{x})$, each multiplied by the horizontal and the vertical components of the velocity basis.

The real and imaginary parts of the basis flow fields for the motion edge are depicted in Figure 8(1-10). Two angular harmonics are shown, each with 4 real-valued flow fields, along with the DC basis functions that encode constant translational velocity. One can see that some of the basis flow fields bear some similarity to nonlinear shear and expansion/compression.

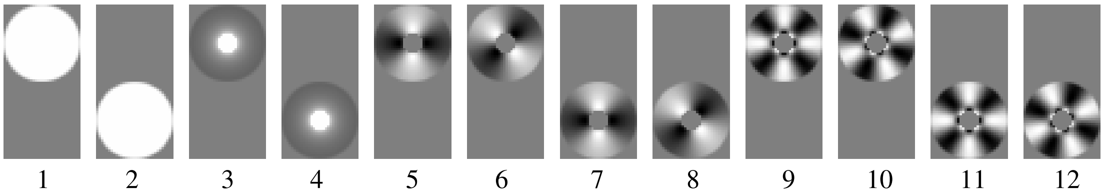


Figure 9: Steerable basis flow fields for motion bars. The DC basis flow fields are equivalent to the first two basis flow fields in the motion edge basis set.

Figure 9(3-12) show the basis flow fields for the motion bar. Here, Figure 9(3-4) encode the basis flow fields for wavenumber $k = 0$, for which only the real-part is nonzero. Figures 9(5-8) and 9(9-12) show the real-valued basis flow fields for wavenumbers $k = 2$ and $k = 4$ respectively.

Note that the basis flow fields for motion edges and bars, based on odd and even wavenumbers respectively, are orthogonal. Thus, when one wants to detect both features, the basis sets can be combined trivially to form a single basis set. We demonstrate them individually, and together in the experiements shown below in Section 5.

Finally, it is also useful to note that these basis flow fields are approximate. With only a small number of wavenumbers the basis flow fields span a relatively crude approximation to a step motion discontinuity. Second, since they approximate only the instantaneous nature of the image velocity field, the bases flow fields do not explicitly model the pixels that are occluded or disoccluded by the moving feature. This results in unmodeled brightness variations at the feature boundaries that must be coped with when estimating the coefficients. Our estimation approach described in Section 4 uses robust statistical techniques for this purpose.

3.2 Learning Motion Models Using PCA

A more general way to construct a model for a particular class of motions is to ”learn” the basis flow fields from a training set that contains representative samples of the class. For simple classes such as motion discontinuities we can generate this training set synthetically. For more complex motions we need to estimate the image motion for training image sequences. Since training is done off-line, we can afford to use a computationally expensive, robust optical flow algorithm [12]. Of course the optical flow technique used to create the training flow fields cannot be expected to provide accurate flow estimates throughout the entire training set. Rather, it is sufficient to assume that the principal structure in the training set is characteristic of the class of motions. Although we use principal components analysis (PCA) here, other methods for dimensionality reduction, such as ICA [6], might be more appropriate for some classes of motion.

Let the training ensemble be a set of p optical flow fields, $\{\mathbf{f}_j(\mathbf{x})\}_{j=1\ldots p}$. For images with s pixels, each flow field contains $2s$ quantities (i.e., the horizontal and vertical elements of the flow at each pixel). For each flow field we place the $2s$ values into a vector by scanning the horizontal components of the flow in standard lexicographic order, followed by the vertical components. This gives us p vectors that become the columns of a $2s \times p$ matrix H. For notational convenience, let $\mathbf{h}_l$, the l^{th} column of H, correspond to the flow field $\mathbf{f}_l(\mathbf{x})$ from the training set. In practice, we take $\mathbf{h}_l$ to be $\mathbf{f}_l(\mathbf{x}) - \bar{\mathbf{f}}(\mathbf{x})$ where $\bar{\mathbf{f}}(\mathbf{x})$ is the mean (flow field) of the training set. With optical flow data, the mean flow is typically close to zero everywhere since, with a large training set, the observed upward motion of the mouth, for example, will roughly "cancel" the downward motion.

PCA can then be used to compute a low-dimensional model for the structure of the flow fields. Toward this end, the singular value decomposition (SVD) of H can be written as

$$H = M \Sigma V^T, \tag{7}$$

where $M = [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p]$ is a $2s \times p$ matrix. The columns, $\mathbf{m}_j$, comprise an orthonormal basis for the range of H, Σ is a $p \times p$ diagonal matrix containing the singular values $\lambda_1, \lambda_2, \ldots, \lambda_p$ sorted in decreasing order along the diagonal, and V^T is a $p \times p$ orthogonal matrix.

We can approximate a given column, $\mathbf{h}_l$, of H by a linear combination of the first n columns in M, associated with the n largest singular values in Σ , that is,

$$\mathbf{h}_l \;\approx\; \tilde{\mathbf{h}}_l \;=\; \sum_{j=1}^{n} c_j \, \mathbf{m}_j \; . \tag{8}$$

where the c_j are the linear coefficients. These n vectors, $\mathbf{m}_j$, comprise a basis for the subspace that approximates the column space of H. Because the basis vectors, $\mathbf{m}_j$, are orthonormal, the optimal approximation to $\mathbf{h}_l$ in the least squares sense is obtained using the coefficients that equal the projection of $\mathbf{h}_l$ onto the basis vectors; i.e., $c_j = \mathbf{h}_l^T \mathbf{m}_j$, for $j = 1, \ldots, n$. The error in the approximation decreases as n increases.

Note that each column of M corresponds to a flow field, as do the columns of H. Thus, if $\mathbf{b}_j(\mathbf{x})$ denotes the flow field that corresponds to $\mathbf{m}_j$, then, from (8), we can approximate each training flow field as a linear combination of the n basis flow fields,

$$\mathbf{f}_l(\mathbf{x}) \;\approx\; \mathbf{u}(\mathbf{x};\, \mathbf{c}) \;=\; \sum_{j=1}^{n} c_j \, \mathbf{b}_j(\mathbf{x}) \; , \tag{9}$$

where $\mathbf{c} = (c_1, \ldots, c_n)^T$, and $\mathbf{u}(\mathbf{x}; \mathbf{c})$ is the approximate flow field.

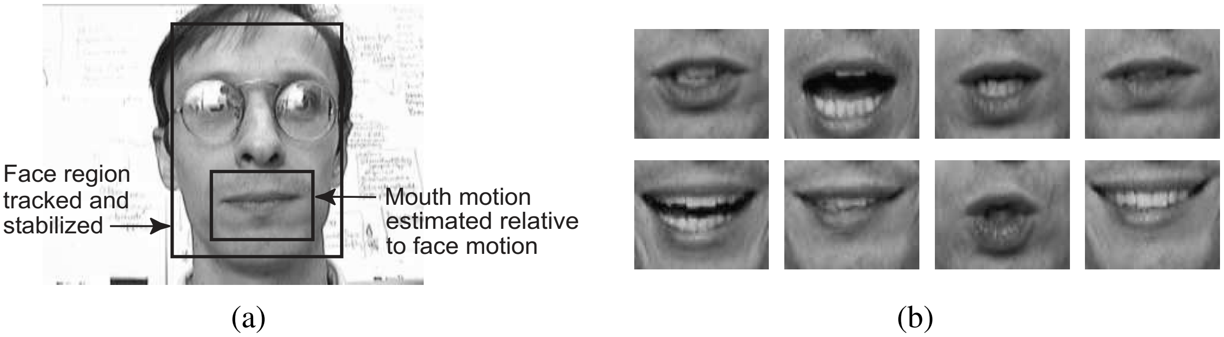


Figure 10: Modeling mouth motion. (a) A subject's head is tracked using a planar model of image motion and the motion of the mouth is estimated relative to this head motion. (b) Example frames from the 3000 image training set of a person saying several words, and changing facial expressions throughout several seconds of video.

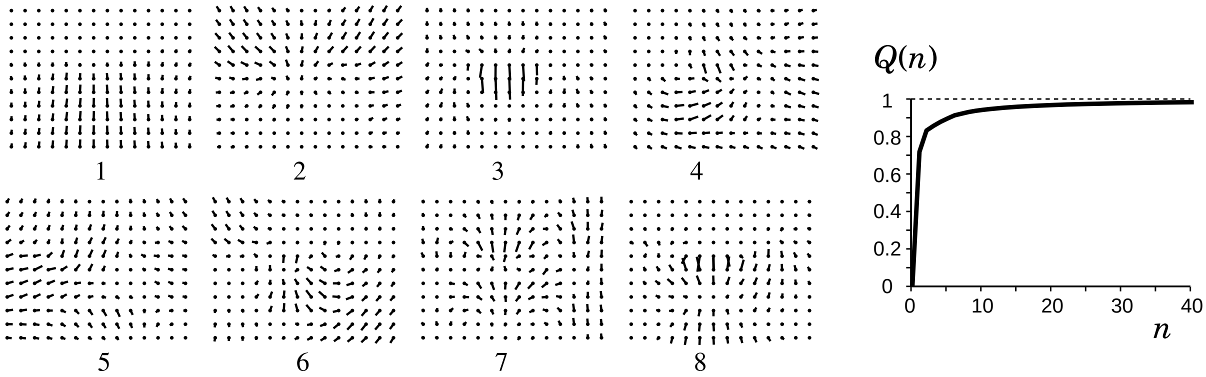


Figure 11: (left) First 8 basis flow fields for non-rigid mouth motion. (right) Fraction of variance from the mouth training ensemble that is captured in the model, as a function of the number of basis flow fields. The first six basis flow fields account for approximately 90% of the variance.

The quality of the approximation provided by the first n columns of M is easily characterized as the fraction of the variance of the training set that is accounted for by the n components:

$$Q(n) = \left(\sum_{j=1}^{n} \lambda_j^2 \right) / \left(\sum_{j=1}^{p} \lambda_j^2 \right) . \tag{10}$$

A good approximation (i.e., when $Q(n)$ approaches 1) is obtained when the singular values λ_j are relatively small for $j > n$. If the singular values rapidly decrease to zero as j increases then $Q(n)$ rapidly increases towards 1, and a low-dimensional linear model provides an accurate approximation to the flow fields in the training set.

3.2.1 Example: Mouth Motion

As an example, we learn a parameterized model of mouth motion for a single speaker. We collected a 3000 image training sequence in which the speaker moved their head, spoke naturally, and made repeated utterances of four test words, namely, “center,” “print,” “track,” and “release.” As illustrated in Figure 10, the subject’s head was tracked and stabilized using a planar motion model to remove the face motion (see [15] for details). This stabilization allows isolation of the mouth region, examples of which are shown in Figure 10. The motion of the mouth region was estimated relative to the head motion using the dense optical flow method described in [12]. This results in a training set of mouth flow fields between consecutive frames.

The mean flow field of this training set is computed and subtracted from each of the training flow fields. The mean motion is nearly zero and accounts for very little of the variation in the training set (1.7%). The modified training flow fields are then used in the SVD computation described above. Since the image motion of the mouth is highly constrained, the optical flow structure in the 3000 training flow fields can be approximated using a small number of principal component flow fields. In this case, 91.4% of the variance in the training set is accounted for by the first seven components (shown in Fig. 11). The first component alone accounts for approximately 71% of the variance. It is interesting to compare this learned basis set with the hand-constructed basis used to model mouth motion in [15]. The hand-constructed basis contained seven flow fields representing affine motion plus a vertical curvature basis to capture mouth curvature. In contrast to the learned basis set, the hand-constructed set accounts for only 77.5% of the variance in the training set.

3.3 Designing Basis Sets

It is also possible to “design” basis sets with particular properties. In many cases one may wish to have a basis that spans affine motion plus some higher-order deformation. This may be true for mouth motion as well as for rigid scenes. For example, we applied the learning method to a training set composed of patches of optical flow (roughly 40×40 sized regions) taken randomly from the Yosemite sequence which contains a rigid scene and a moving camera [3]. The first six basis flows accounted for 99.75% of the variance, and they appear to span affine motions. To compare this with an exact affine model, we projected all affine motion out of the training flow fields, and then performed PCA on the residual flows fields. The affine model accounts for 99.68% of the variance suggesting that, for this sequence and others like it, a local affine model is sufficient.

To construct a basis to explicitly represent affine motion plus some non-rigid deformation from a training set, we first construct an orthonormal set of affine basis flows,

$$\mathbf{u}(\mathbf{x};\mathbf{c}) = \begin{bmatrix} u(\mathbf{x};\mathbf{c}) \\ v(\mathbf{x};\mathbf{c}) \end{bmatrix} = \begin{bmatrix} c_1 + c_2 x + c_3 y \\ c_4 + c_5 x + c_6 y \end{bmatrix},$$

the affine basis flows $\mathbf{b}_j(\mathbf{x})$ of which are illustrated in Figure 1. We project affine structure out of the training set using a Gram-Schmidt procedure [31], and then apply PCA to the remaining flow fields. More precisely, in vector form, let $\{\mathbf{m}_j\}_{j=1\ldots6}$ denote an orthonormal basis for affine motion, and let the original ensemble of training flow fields be expressed as $\{\mathbf{h}_j\}_{j=1\ldots p}$. The magnitude of the projection of $\mathbf{h}_l$ onto the affine basis vectors $\mathbf{m}_j$ is given by $c_{lj} = \mathbf{h}_l^T \mathbf{m}_j$. Then, affine structure can be subtracted out of $\mathbf{h}_l$ to produce the new training flow field, $\mathbf{h}_l^*$, given by

$$\mathbf{h}_l^* \; = \; \mathbf{h}_l - \sum_{j=1}^{6} c_{lj} \, \mathbf{m}_j \; . \tag{11}$$

Finally, we perform PCA on the new training set, each flow field of which is now orthogonal to the affine basis set. We choose the first n basis flows from this deformation basis and then construct our basis set with $6 + n$ flow fields in which the first six represent affine motion. We now have a basis that is guaranteed to represent affine motion plus some learned deformation from affine. This process can be also be applied to orthogonal basis sets other than affine.

4 Direct Estimation of Model Coefficients

Given a basis set of flow fields for a particular motion class, we wish to estimate the model coefficients from an image sequence. We then wish to use these coefficients to detect instances of the motion classes. To detect features in static images using linear parameterized feature models (e.g., [42]), the model coefficients are obtained by convolving the image with the basis images. With motion models we cannot take the same approach because the motion field is unknown. One could first estimate the dense flow field using generic smoothness constraints and then filter the result. However, the strong constraints provided by parameterized models have been shown to produce more accurate and stable estimates of image motion than generic dense flow models [38]. Therefore we apply our new motion models in the same way that affine models have been used successfully in the past; we make the assumption of brightness constancy and estimate the linear coefficients directly from the spatial and temporal image derivatives.

More formally, within an image region, R, we wish to find the linear coefficients $\mathbf{c}$ of a parameterized motion model that satisfy the brightness constancy assumption,

$$I(\mathbf{x},\, t+1) - I(\mathbf{x} - \mathbf{u}(\mathbf{x};\mathbf{c}),\, t) \; = \; 0 \quad \forall \mathbf{x} \in R \, , \tag{12}$$

where $\mathbf{u}(\mathbf{x};\mathbf{c})$ is given by (1). Equation (12) states that the image, I, at frame $t + 1$ is a warped version of the image at time t.

In order to estimate the model coefficients we minimize the following objective function

$$E(\mathbf{c}) = \sum_{\mathbf{x}\in R} \rho(I(\mathbf{x},\, t+1) - I(\mathbf{x} - \mathbf{u}(\mathbf{x};\mathbf{c}),\, t),\; \sigma)\,. \tag{13}$$

Here, σ is a scale parameter and $\rho(\cdot, \sigma)$ is a robust error function applied to the residual error

$$\Delta I(\mathbf{x};\, \mathbf{c}) = I(\mathbf{x}, t+1) - I(\mathbf{x} - \mathbf{u}(\mathbf{x};\mathbf{c}), t)\,. \tag{14}$$

Large residual errors may be caused by changes in image structure that are not accounted for by the learned flow model. Because of the discontinuous nature of the motion models used here and the expected violations of brightness constancy, it is important that the estimator be robust with respect to these “outliers”. For the experiments below we take $\rho(\cdot, \sigma)$ to be

$$\rho(r, \sigma) \;=\; \frac{r^2}{\sigma^2 + r^2}\,,$$

which was proposed in [30] and used successfully for flow estimation in [12]. The parameter, σ, controls the shape of $\rho(\cdot, \sigma)$ to minimize the influence of large residual errors on the solution.

Equation (13) can be minimized in a number of ways including coordinate descent [12], random sampling [2], or iteratively reweighted least squares [1, 32]. Here we use a coarse-to-fine, iterative, coordinate descent method. To formulate an iterative method to minimize (13), it is convenient to first rewrite the model coefficient vector in terms of an initial guess $\mathbf{c}$ and an update $\delta\mathbf{c}$. This allows us to rewrite (13) as

$$E(\delta\mathbf{c};\mathbf{c}) = \sum_{\mathbf{x}\in R} \rho(I(\mathbf{x}, t+1) - I(\mathbf{x} - \mathbf{u}(\mathbf{x};\, \mathbf{c} + \delta\mathbf{c}), t),\; \sigma). \tag{15}$$

Given an estimate, $\mathbf{c}$, of the motion coefficients (initially zero), the goal is to estimate the update, $\delta\mathbf{c}$, that minimizes (15); $\mathbf{c}$ then becomes $\mathbf{c} + \delta\mathbf{c}$. To minimize (15) we first approximate it by linearizing the residual, $\Delta I(\mathbf{x};\, \mathbf{c} + \delta\mathbf{c})$, with respect to the update vector $\delta\mathbf{c}$ to give

$$\tilde{E}(\delta\mathbf{c};\mathbf{c}) \;=\; \sum_{\mathbf{x}\in R} \rho(\mathbf{u}(\mathbf{x};\, \delta\mathbf{c})^T \vec{\nabla} I(\mathbf{x} - \mathbf{u}(\mathbf{x};\, \mathbf{c}), t) + \Delta I(\mathbf{x};\mathbf{c}),\; \sigma), \tag{16}$$

where $\vec{\nabla} I(\mathbf{x} - \mathbf{u}(\mathbf{x};\mathbf{c}), t)$ denotes the spatial image gradient at time t, warped by the current motion estimate $\mathbf{u}(\mathbf{x};\mathbf{c})$ using bilinear interpolation. Note that the brightness constancy assumption has been approximated by an optical flow constraint equation that is linear in $\delta\mathbf{c}$. Finally, note that in minimizing (16), the search algorithm described below typically generates small update vectors, $\delta\mathbf{c}$. Because the objective function in (16) satisfies $\tilde{E}(\delta\mathbf{c};\mathbf{c}) = E(\delta\mathbf{c};\mathbf{c}) + O(||\delta\mathbf{c}||^2)$, the approximation error vanishes as the update, $\delta\mathbf{c}$, is reduced to zero.

It is also worth noting that the gradient term in (16) does not depend on $\delta\mathbf{c}$. This avoids the need to rewarp the image and recompute the image gradient at each step of the coordinate descent. In fact, the image gradient in (16) can be pre-multiplied by the basis flows fields since these quantities will not change during the minimization of $\tilde{E}(\delta\mathbf{c};\ \mathbf{c})$. Hager and Belhumeur [32] used this fact for real-time affine tracking.

To minimize (15) we use a coarse-to-fine control strategy with a continuation method that is an extension of that used by Black and Anandan [12] for estimating optical flow with affine and planar motion models. We construct a Gaussian pyramid for the images at time t and $t+1$. The motion coefficients, $\mathbf{c}_l$, determined at a coarse scale, l, are used in the minimization at the next finer scale, $l+1$. In particular, the motion coefficients, $\mathbf{c}_l + \delta\mathbf{c}_l$, from the coarse level are first multiplied by a factor of two (to reflect the doubling of the image size at level $l+1$) to produce the initial guess $\mathbf{c}_{l+1}$ at level $l+1$. These coefficients are then used in (16) to warp the image at time $t+1$ towards the image at time t at level $l+1$, from which $\tilde{E}(\delta\mathbf{c}_{l+1}; \mathbf{c}_{l+1})$ is minimized to compute the next update $\delta\mathbf{c}_{l+1}$.

The basis flow fields at a coarse scale are smoothed, subsampled versions of the basis flows at the next finer scale. These coarse-scale basis vectors may deviate slightly from orthogonality but this is not significant given our optimization scheme. We also find that it is important for the stability of the optimization to use fewer basis flow fields at the coarsest levels, increasing the number of basis flow fields as the estimation proceeds from coarse to fine. The basis flow fields to be used at the coarsest levels are those that correspond to the majority of the energy in the training set (for domain-specific models) or in the approximation to the model feature (for motion features). The flows fields used at the coarse levels are typically smoother (i.e., lower wavenumbers), and therefore they do not alias or lose significant signal power when they are subsampled.

At each scale, given a starting estimate $\mathbf{u}(\mathbf{x}; \mathbf{c}_l)$, we minimize $\tilde{E}(\delta\mathbf{c}; \mathbf{c}_l)$ in (16) using a coordinate descent procedure. The partial derivatives of $\tilde{E}(\delta\mathbf{c}; \mathbf{c}_l)$ with respect to $\delta\mathbf{c}$ are straightforward to derive. When the update vector is complex-valued, as it is with our motion feature models, we differentiate $\tilde{E}(\delta\mathbf{c}; \mathbf{c}_l)$ with respect to both the real and imaginary parts of $\delta\mathbf{c}$. To deal with the non-convexity of the objective function, the robust scale parameter, σ, is initially set to a large value and then slowly reduced. For the experiments below, σ is lowered from $25\sqrt{2}$ to $15\sqrt{2}$ by a factor of 0.95 at each iteration. Upon completion of a fixed number of descent steps (or when a convergence criterion is met), the new estimate for the flow coefficients is taken to be $\mathbf{c}_l + \delta\mathbf{c}$.

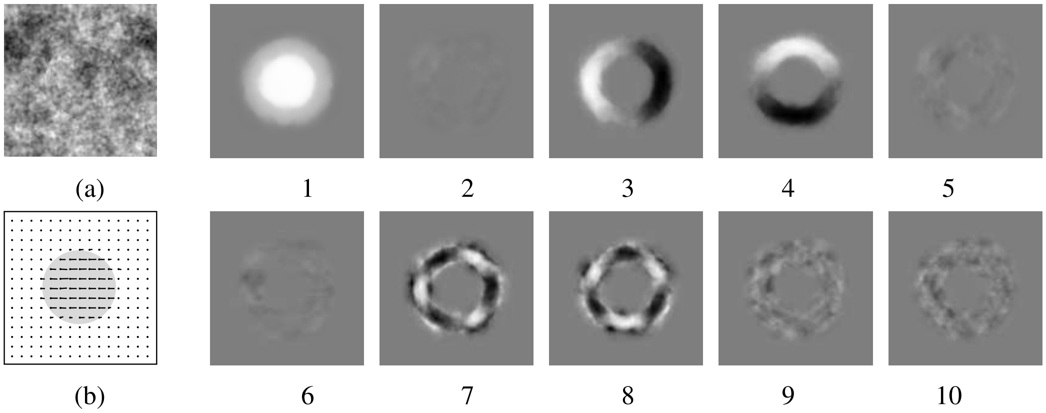


Figure 12: Translating Disk. (a) Image of the disk and background (same random texture). (b) Estimated flow field superimposed on the disk. (1-10) Recovered coefficient images for the motion edge basis set. For display the responses of coefficients corresponding to each wavenumber are scaled independently to maximize the range of gray levels shown.

4.1 Translating Disk Example

To illustrate the estimation of the model coefficients and the detection of motion discontinuities, we constructed a synthetic sequence of a disk translating across a stationary background (Figure 12(a)). Both the disk and the background had similar fractal textures, so the boundary of the stationary disk is hard to see in a single image. The image size was 128×128, the disk was 60 pixels wide, and the basis flow fields were 32 pixels wide.

Optical flow estimation with a motion feature basis produces a set of coefficients for each neighborhood in the image. Here, the model coefficients were estimated from neighborhoods centered at each pixel (so long as neighborhoods did not overlap the image boundary). This yields coefficient values at each pixel that can be viewed as images (i.e., two real-valued images for each complex-valued coefficient). Figure 12(1-10) shows the real-valued coefficients that correspond to the basis flow fields for the motion edge model in Figure 8(1-10). Figure 12(1,2) depicts coefficients for horizontal and vertical translation; one can infer the horizontal velocity of the disk. Figure 12(3-10) corresponds to basis flow fields with horizontal and vertical motion at wavenumbers 1 and 3. In these images one can see the dependence of coefficient values on the wavenumber, the orientation of the motion edge, and the direction of the velocity difference. As described below, it is the structure in these coefficients that facilitates the detection of the motion edges.

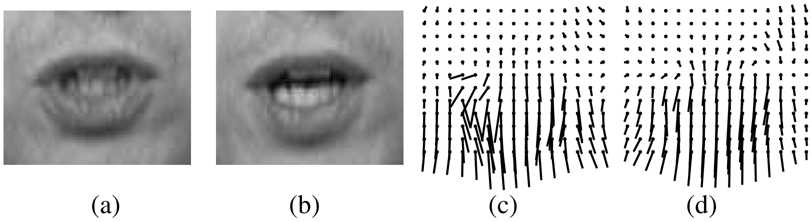


Figure 13: Estimating mouth motion. (a,b) mouth regions of two consecutive images of a person speaking. (c) Flow field estimated using dense optical flow method. (d) Flow field estimated using the learned model with 6 basis flow fields.

4.2 Human Mouth Motion Example

Estimation of human mouth motion is challenging due to non-rigidity, self occlusion, and high velocities relative to standard video frame rates. Consider the two consecutive frames from a video sequence shown in Figure 13. Note the large deformations of the lips, and the changes in appearance caused by the mouth opening. This makes optical flow estimation with standard dense flow methods difficult. For example, Figure 13(c) shows a flow field estimated with a robust dense method [12]. Some of the flow vectors differ widely from their neighbors. In comparison, the flow field estimated with the learned model is constrained to be a type of mouth motion, which yields the smoother flow field in Figure 13(d). Of course, we cannot say which of these two flow fields is "better" in this context; each minimizes a measure of brightness constancy. In the following section we will be concerned with recognition, and in that scenario we wish to constrain the estimation to valid mouth motions such as that represented by Figure 13(d).

5 Detection of Parameterized Motion Models

From the estimated linear model coefficients, we are interested in detecting and recognizing types of motion. We first consider the detection of motion edges and bars. Following that, we examine the recognition of time-varying motions in the domains of mouths and people walking.

5.1 Motion Feature Detection

Given the estimated linear coefficients, **c**, we wish to detect occurrences of motion edges and motion bars, and to estimate their parameters, namely, the two components of the mean velocity $\mathbf{u}_t = (u_t, v_t)$, the two components of the velocity change $\Delta\mathbf{u} = (\Delta u, \Delta v)^T$, and the feature orientation θ. The linear parameterized models for motion edges and bars were designed so that any motion edge or bar can be well approximated by some linear combination of the basis flow fields. But not all linear combinations of the basis flow fields correspond to valid motion features. Rather, the motions features lie on a lower-dimensional, nonlinear manifold within the subspace spanned by the basis flow fields. In [42], the detection problem for static features was addressed by taking tens of thousands of example features and projecting them onto the basis set. These training coefficients provided a dense sampling of a manifold in the subspace spanned by the basis vectors. Given the coefficients corresponding to an image feature, the closest point on the manifold was found. If an estimated coefficient vector **c** lies sufficiently close to the manifold then the parameters of the nearest training example were taken to be the model parameters. Alternatively, one could interpolate model parameters over the manifold [20]. This approach to detection is appropriate for complex features where no underlying model is available (e.g., the motion of human mouths [16]). By comparison, the underlying models of motion features are relatively simple. In these cases we therefore solve for the feature parameters directly from the linear coefficients.

5.1.1 Nonlinear Least-Squares Estimation

Given an estimated flow field, $\mathbf{u}(\mathbf{x}; \mathbf{c})$, obtained with the robust iterative method outlined in Section 4, we wish to detect the presence of a motion feature and to estimate its parameters $(\mathbf{u}_t, \Delta\mathbf{u}, \theta)$. That is, we wish to estimate the parameters that produce the idealized flow field, $\mathbf{f}(\mathbf{x};\, \mathbf{u}_t, \Delta\mathbf{u}, \theta)$, that is closest to the estimated flow field, $\mathbf{u}(\mathbf{x};\, \mathbf{c})$. We also want to decide whether the model is a sufficiently good fit to allow us to infer the presense of the feature. Because there are five independent feature parameters, the orthogonal projection of $\mathbf{f}(\mathbf{x};\, \mathbf{u}_t, \Delta\mathbf{u}, \theta)$ onto the basis flow fields lie on a 5-dimensional nonlinear manifold. It suffices to find the flow field on this manifold that is closest to $\mathbf{u}(\mathbf{x};\, \mathbf{c})$.

Let $\mathbf{u}^m(\mathbf{x};\, \mathbf{u}_t, \Delta\mathbf{u}, \theta)$ denote the projection of idealized flow fields $\mathbf{f}(\mathbf{x};\, \mathbf{u}_t, \Delta\mathbf{u}, \theta)$, onto the linear subspace. Using the form of the basis flow fields in (3), (4) and (6), one can show that $\mathbf{u}^m(\mathbf{x};\, \mathbf{u}_t, \Delta\mathbf{u}, \theta)$ has the form

$$\mathbf{u}^m(\mathbf{x};\, \mathbf{u}_t, \Delta\mathbf{u}, \theta) \;=\; \mathbf{u}_t + \Re\left[\sum_{k\in K} \sigma_k e^{-ik\theta}\left(\Delta u\, \mathbf{b}_k^h(\mathbf{x}) + \Delta v\, \mathbf{b}_k^v(\mathbf{x})\right)\right] \;. \tag{17}$$

Then, for a region R, we seek the five parameters $(\mathbf{u}_t, \Delta\mathbf{u}, \theta)$ that minimize

$$\sum_{\mathbf{x}\in R} \|\; \mathbf{u}^m(\mathbf{x};\, \mathbf{u}_t,\, \Delta\mathbf{u},\, \theta) \;-\; \mathbf{u}(\mathbf{x}; \mathbf{c}) \;\|^2 \;. \tag{18}$$

To formulate the solution it is convenient to explicitly express the estimated flow field, $\mathbf{u}(\mathbf{x}; \mathbf{c})$, using the same notation for the basis flow fields:

$$\mathbf{u}(\mathbf{x};\mathbf{c}) \;=\; \mathbf{u}_{dc} \,+\, \Re\left[\sum_{k\in K}(\alpha_k\mathbf{b}_k^h(\mathbf{x}) \,+\, \beta_k\mathbf{b}_k^v(\mathbf{x}))\right]\,, \tag{19}$$

where $\mathbf{u}_{dc}$ is a 2-vector that corresponds to the coefficients associated with the DC basis flow fields, and α_k and β_k are the estimated (complex-valued) coefficients that correspond to the horizontal and vertical components of the basis flow fields with wavenumber k.

The basis in which both $\mathbf{u}^m(\mathbf{x};\,\mathbf{u}_t,\Delta\mathbf{u},\theta)$ and $\mathbf{u}(\mathbf{x};\mathbf{c})$ are expressed (i.e., $\{\mathbf{b}_k^h(\mathbf{x}),\mathbf{b}_k^v(\mathbf{x})\}_{k\in K}$) is orthogonal. Therefore, to minimize (18) it suffices to find the feature parameters that minimize the sum of squared differences between model coefficients and the estimated coefficients. Thus, the translational velocity is given directly by $\mathbf{u}_t = \mathbf{u}_{dc}$. The remaining parameters, $\Delta\mathbf{u}$ and θ, are found by minimizing

$$E_m(\Delta\mathbf{u},\theta) \;=\; \sum_{k\in K} \|\; (\alpha_k,\beta_k) - \sigma_k e^{-ik\theta}(\Delta u,\Delta v)\;\|^2 \quad , \tag{20}$$

given a sufficiently good initial guess.

The least-squares minimization enforces two constraints on the feature parameters. First, the velocity difference, $\Delta\mathbf{u}$, must be consistent over all angular harmonics. Second, the orientation of the motion feature, θ, must be consistent across all of the angular harmonics and both components of flow. The constraint on $\Delta\mathbf{u}$ is related to the magnitudes of the complex-valued model coefficients, while the constraint on θ concerns their phase values. To obtain an initial guess for minimizing $E_m(\Delta\mathbf{u},\theta)$, we first decouple these constraints. This provides a suboptimal, yet direct, method for estimating θ and $\Delta\mathbf{u}$.

5.1.2 Direct Estimation of Velocity Change.

To formulate a constraint on $\Delta\mathbf{u} = (\Delta u,\Delta v)^T$ it is convenient to collect the complex-valued coefficients of the model flow field in (17) into an outer product matrix

$$\begin{aligned} M \;&=\; \Delta\mathbf{u}\,\left(\sigma_{k_1}\,e^{-ik_1\theta}\,,...,\;\sigma_{k_n}\,e^{-ik_n\theta}\right) && (21)\\ &=\; \begin{pmatrix} \Delta u\,\sigma_{k_1}\,e^{-ik_1\theta} & ... & \Delta u\,\sigma_{k_n}\,e^{-ik_n\theta} \\ \Delta v\,\sigma_{k_1}\,e^{-ik_1\theta} & ... & \Delta v\,\sigma_{k_n}\,e^{-ik_n\theta}\end{pmatrix}\,, && (22)\end{aligned}$$

where n is the number of angular harmonics in K, and for $1\le j\le n$, let k_j denote the wavenumbers in K with weights σ_k. The top row of M contains the model coefficients for the horizontal components of the model velocity field. The second row of M contains the model coefficients for the vertical components of $\mathbf{u}^m(\mathbf{x};\,\mathbf{u}_t,\Delta\mathbf{u},\theta)$.

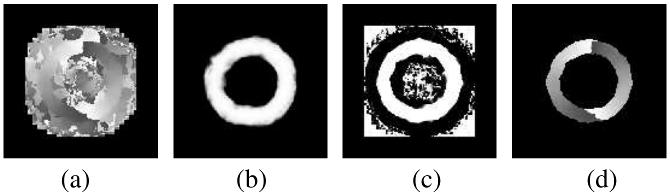


Figure 14: Translating Disk. (a) Raw orientation estimates from direct method. Black pixels indicate that no estimate was possible. Intensity varies from white to dark grey as orientation goes from $\pi/2$ to $-\pi/2$. (b,c) Confidence measures from velocity constraint and orientation constraint. (d) orientation estimates masked by joint confidence measure.

To obtain a set of transformed model coefficients that depend solely on $\Delta\mathbf{u}$, we then construct $A = M\,M^{*T}$, where M^{*T} is the conjugate transpose of M. Because of the separability of the model coefficients with respect to $\Delta\mathbf{u}$ and θ, as shown in (21), A reduces to $(\sigma_{k_1}^2 + \ldots + \sigma_{k_n}^2)\,\Delta\mathbf{u}\,\Delta\mathbf{u}^t$, the components of which are independent of θ. For example, in the case of the motion edge, with the estimated coefficients α_1, α_3, β_1 and β_3, let

$$\tilde{M} \;=\; \begin{pmatrix} \alpha_1 & \alpha_3 \\ \beta_1 & \beta_3 \end{pmatrix} \,, \quad \tilde{A} \;=\; \tilde{M}\,\tilde{M}^{*T} \,. \tag{23}$$

If the estimated flow field, $\mathbf{u}(\mathbf{x};\mathbf{c})$, were on the feature manifold, then the singular vector associated with the largest singular value, e_1, of $\tilde{A}$ should give the direction of the velocity. Thus, the estimate of the velocity change, $\Delta\tilde{\mathbf{u}}_0$, is obtained by scaling this singular vector by $\sqrt{e_1/(\sigma_1^2+\sigma_3^2)}$.

Moreover, if the estimated coefficients lie on the feature manifold, then the rank of $\tilde{A}$ in (23) should be 1. This suggests that we can use the ratio of the singular values, $e_1 \geq e_2$, to determine the quality of the model fit. A measure of the consistency of the estimated coefficients with the model is given by $r = (e_2 + a)/e_1$. Here, r is close to zero where the coefficients satisfy the constraint and large otherwise; a is a small constant that helps to avoid instability when both singular values are extremely small. We use $C_v = \exp[-r^2]$ as a confidence measure derived from this constraint, an image of which is shown for the disk sequence in Figure 14(b).

5.1.3 Direct Estimation of Spatial Orientation.

Once the velocity parameters have been estimated, we can use the initial estimate, $\Delta\tilde{\mathbf{u}}_0$, along with the matrix $\tilde{M}$ in (23) to obtain a set of transformed measurements that, according to the model, depend only on the spatial structure of the motion feature. In particular, given the true velocity change $\Delta\mathbf{u}$, it is easy to show from (21) that the product $\Delta\mathbf{u}^T M$ has the form

$$\Delta\mathbf{u}^T M \;=\; ||\Delta\mathbf{u}||^2\,\left(\sigma_{k_1}\,e^{-i\,k_1\,\theta}\,, \ldots,\; \sigma_{k_n}\,e^{-i\,k_n\,\theta}\right) \,. \tag{24}$$

From this it is clear that the orientation θ is available in the phase of the elements of $\Delta\mathbf{u}^T M$.

To obtain an initial orientation estimate $\tilde{\theta}_0$, we form the product $\mathbf{z} = \Delta\tilde{\mathbf{u}}_0^T \tilde{M}$, using the estimated velocity change $\Delta\tilde{\mathbf{u}}_0$ and the matrix of estimated coefficients $\tilde{M}$. To obtain $\tilde{\theta}_0$ from $\mathbf{z}$ we divide the phase of each component of $\mathbf{z}$ according to its corresponding wavenumber (taking phase wrapping into account), and take their average. According to the model, ignoring noise, the resulting phase values should then be equal, and computing their average yields the orientation estimation $\tilde{\theta}_0$. Figure 14(a) shows the $\tilde{\theta}_0$ as a function of image position for the Disk Sequence.

The variance of the normalized phases also gives us a measure of how well the estimated coefficients satisfy the model. For the edge model, where only two harmonics are used, we expect that $\Delta\phi = \phi_1 - \phi_3/3 = 0$ where ϕ_k is the phase of the component of $\mathbf{z}$ at wavenumber k. As a simple confidence measure for the quality of the model fit, Figure 14(c) shows $C_\theta = \exp(-\Delta\phi^2)$ for the disk sequence.

5.1.4 Least-Squares Estimation and Detection

The direct method provides initial estimates of orientation $\tilde{\theta}_0$ and velocity change $\Delta\tilde{\mathbf{u}}_0$, with confidence measures, C_v and C_θ. We find that these estimates are usually close to the least-squares estimates we seek in (20). The confidence measures can be used to reject coefficient vectors that are far from the feature manifold. For example, Figure 14(d) shows orientation estimates where $C_v\, C_\theta > 0.1$.

Given these initial estimates, $\tilde{\theta}_0$ and $\Delta\tilde{\mathbf{u}}_0$, we use a gradient descent procedure to find $\tilde{\theta}$ and $\Delta\tilde{\mathbf{u}}$ that minimize $E_m(\Delta\mathbf{u}, \theta)$ in (20). Feature detection is then based on the squared error at the minima divided by the energy in the subspace coefficients, i.e., $P = \sum_k (|\alpha_k|^2 + |\beta_k|^2)$. We find that the reliability of the detection usually improves as P increases. To exploit this, we use a simple confidence measure of the form

$$C = c(P)\, e^{-E/P} \; . \tag{25}$$

where $c(P) = e^{-\kappa/P}$ and κ is a positive scalar that depends on noise levels. In this way, as P decreases the relative error E/P must decrease for our confidence measure, C, to remain constant.

Figure 15(a) shows the confidence measure C given the least-squares estimates of the motion edge parameters, where $\kappa = 40$. Figure 15(b) shows locations at which $C > 0.8$, which are centered about the location of the edge. The final three images in Figures 15 show optimal estimates of $\tilde{\theta}$, $\Delta\tilde{\mathbf{u}}$ and $\Delta\tilde{v}$ where $C > 0.8$. Note how the velocity differences clearly indicate the occluding and disoccluding sides of the moving disk. For these pixels, the mean error in θ was $0.12°$ with a standard deviation of $5.6°$. The mean error in Δu was -0.25 pixels/frame with a standard deviation

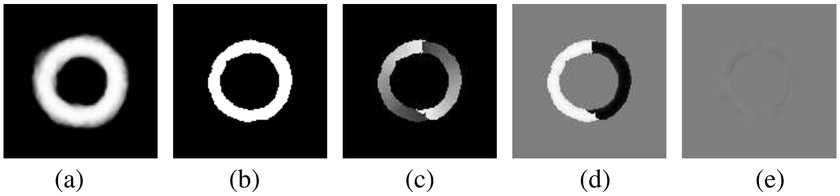


Figure 15: Translating Disk. (a) Confidence measure C, derived from squared error of least-squares fit. (b) Locations where $C > 0.8$ are white. (c-e) Estimates of θ, Δu, Δv where $C > 0.8$.

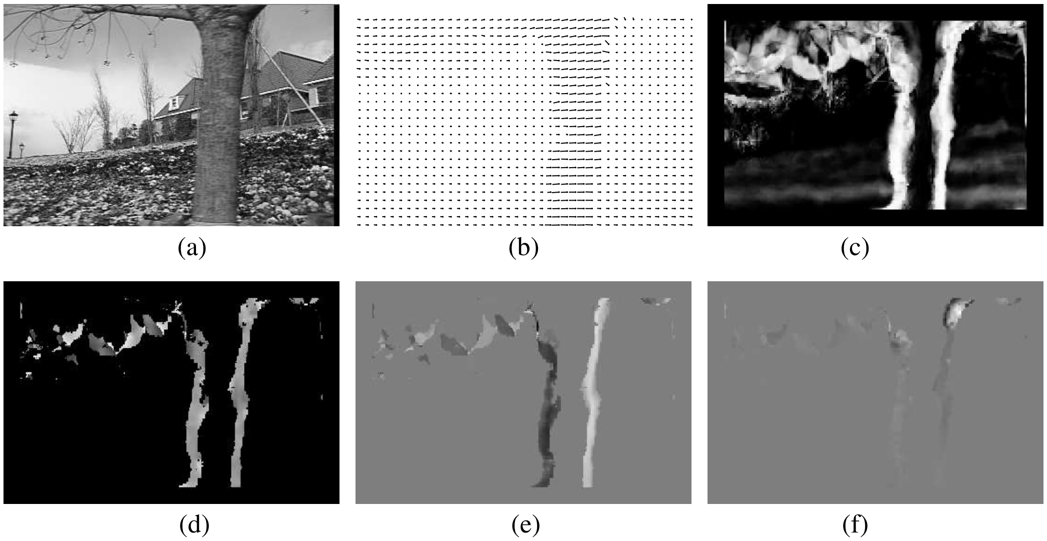


Figure 16: Flower Garden Sequence. (a) Frame 1 of sequence. (b) Recovered flow. (c) Confidence measure C. (d-f) Optimal estimates of θ, Δu and Δv where $C > 0.8$.

of 0.19. Errors in Δv were insignificant by comparison.

Figure 16 shows the detection and estimation of motion edges from the flower garden sequence (an outdoor sequence with translational camera motion). The velocity differences at the tree boundary in the flower garden sequence are as large as 7 pixels/frame. The sign of the velocity change in Figure 16(e) clearly shows the occluding and disoccluding sides of the tree. The orientation estimates along tree are nearly vertical, as indicated by the grey pixels in 16(d).

Further experimental results can be found in [17], where the motion edge detector described here is used to initialize a probabilistic particle filter for detecting and tracking motion discontinuities. The probabilistic method uses a higher dimensional, nonlinear model, and it relies on the linear model and the detection method described here to provide it with rough estimates of the parameters of the nonlinear model. More precisely, it uses these estimates to constrain its search to those regions of the parameter space that have the greatest probability of containing the maximum likelihood estimates of the velocities, orientation, and position of the motion boundary.

To explore the detection of moving bars, we formed a basis set by combining the motion bar

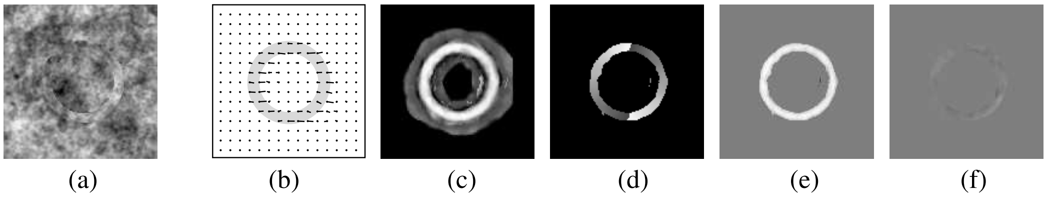


Figure 17: Translating Annulus. (a) Frame 1 of sequence. (b) Recovered flow. (c) Confidence measure C from LS minimization. (d-f) Optimal estimates of θ, Δu and Δv where $C > 0.65$.

basis flow fields (Fig. 9) with the basis containing the translation and motion edge models (Fig. 8). The resulting set of 20 flow fields is orthogonal because the edge and bar basis functions contain odd and even numbered harmonics respectively. In this case, the support of the basis flow fields is a circular window 32-pixels wide. The bar was in the center of the region, with a width of 8 pixels.

To test the detection of moving bars with this basis, we first constructed a synthetic sequence of an annulus (width of 8 pixels) translating across a stationary background to the right at 2 pixels/frame (see Figure 17(a-b)). The detection procedure is identical to that for moving edges. Figure 17(c) shows C at the least-squares minimum, with $\kappa = 50$. The remaining images, Figures 17(d-f) show the optimal estimates of θ and $\Delta\mathbf{u}$ at pixels where $C > 0.65$. The fits with the motion bar model are noisier than those with the edge model, and we therefore use a more liberal threshold to display the results. For these pixels, the mean error in θ was $1.0°$ with a standard deviation of $11.8°$. The mean error in Δu was -0.39 pixels/frame with a standard deviation of 0.31. Note that in these experiments, although the models for the edge and bar are straight, we are testing them with curved edges and bars. The results illustrate how these simple models generalize.

Finally Figure 18 shows the detection and estimation of the motion bars in outdoor image sequence taken with a hand-held camcorder. This sequence depicts a narrow tree (about 6 pixels wide) moving in front of the ground plane. The difference in velocity is predominantly horizontal, while the orientation of the tree is nearly vertical. Where the change in velocity between the tree and ground is sufficiently large, towards the upper part of the tree, the moving bar is detected well.

5.2 Domain-Specific Experimental Results

In the case of generic motion features it was possible to evaluate the performance of the basis set with respect to an idealized optical flow model. In the case of domain-specific models such as human mouths, in which the basis set is constructed from examples, no such idealized models exist. Therefore, to evaluate the accuracy and stability of the motion estimated with these models we consider the use of the recovered coefficients for the task of recognition of motion events. This

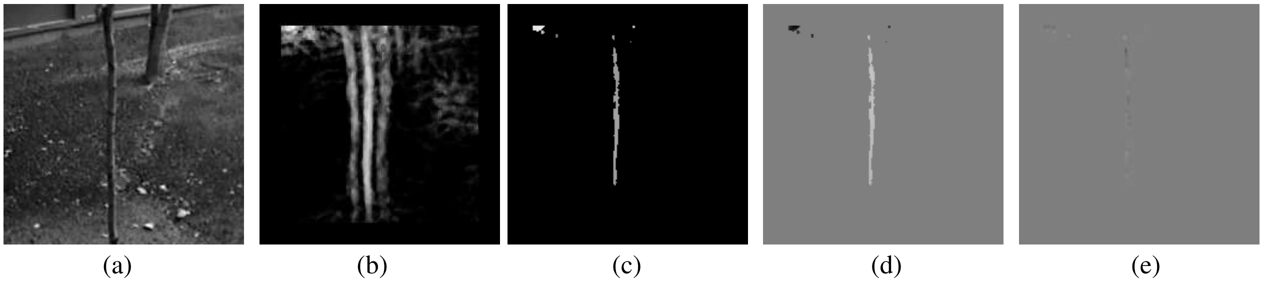


Figure 18: Small Tree Sequence. (a) Frame 1 of sequence. (b) Confidence measure C. (c-e) Optimal estimates of θ, Δu and Δv where $C > 0.65$.

will help demonstrate that the recovered motion is a reasonable representation of the true optical flow.

Below we consider domain specific models for human mouths and legs and illustrate the use of the recovered coefficients with examples of lip reading and walking gait detection. These recognition tasks will necessitate the construction of temporal models that capture the evolution of the coefficients over a number of frames.

5.2.1 Non-Rigid Human Mouth Motion

Black and Yacoob [15] described a method for recognizing human facial expressions from the coefficients of a parameterized motion model. They modeled the face as a plane and used its motion to stabilize the image sequence. The motion of the eyebrows and mouth were estimated relative to this stabilized face using a seven parameter model (affine plus a vertical curvature term). While this hand-coded model captures sufficient information about feature deformation to allow recognition of facial expressions, it it does not capture the variability of human mouths observed in natural speech.

To test the accuracy and stability of the learned model we apply it the problem of lip reading for a simple user interface called the Perceptual Browser [10]. The Perceptual Browser tracks the head of a human sitting in front of a computer display and uses the subject's vertical head motion to control the scrolling of a window such as a web browser.

We explore the addition of four mouth "gestures" to control the behavior of the browser:

- **Center**: mark the current head position as the "neutral" position.
- **Track**: start tracking the head. In this mode the head acts like a "joystick" and deviations from the neutral position cause the page to scroll.
- **Release**: stop head tracking. Head motions no longer cause scrolling.

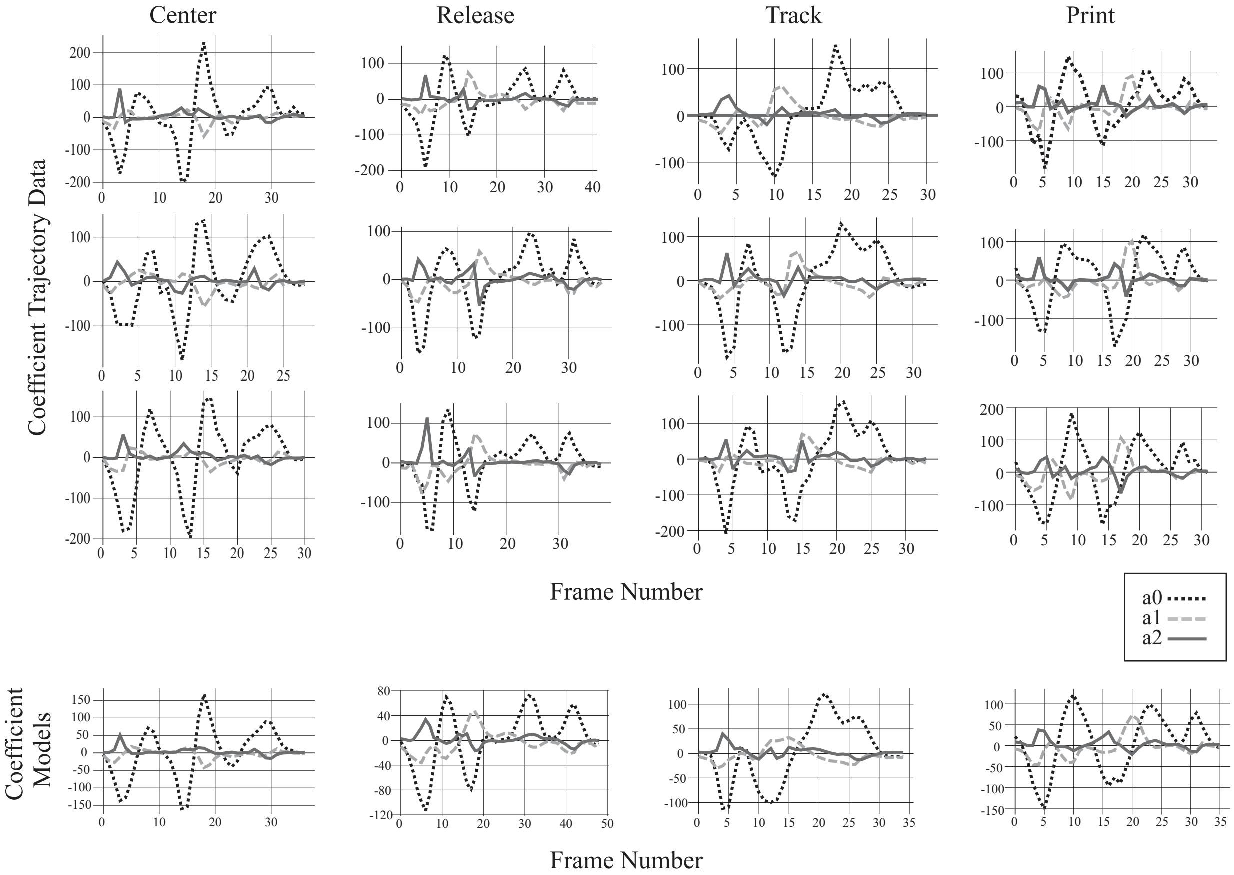


Figure 19: (Top) Example training sequences of mouth motion coefficients. (bottom) Temporal models are constructed from the examples, to produce model coefficient trajectories.

- **Print**: print the contents of the current page.

We think of these as “gestures” in that the user does not need to vocalize the commands but simply makes the mouth motion associated with the words.

For training we used the 3000 image sequence described in Section 3.2.1 that contains the test utterances as well as other natural speech and facial expressions. The head location was assumed to be known in the first frame of the sequence and the head was tracked thereafter using a planar motion model. The mouth location relative to the head was also known. The first three basis vectors (with the largest eigenvalues) were used for estimating the motion; as discussed in Section 3.2.1, the mean flow $\bar{\mathbf{f}}$ was not used as it accounts for significantly less variance than the first three basis vectors.

Figure 19 shows the coefficients estimated for the utterances in the training sequence by minimizing (13). Note that the temporal trajectories for the coefficients of a single utterance produce curves that are very similar while the trajectories for different utterances are distinct. Trajectory

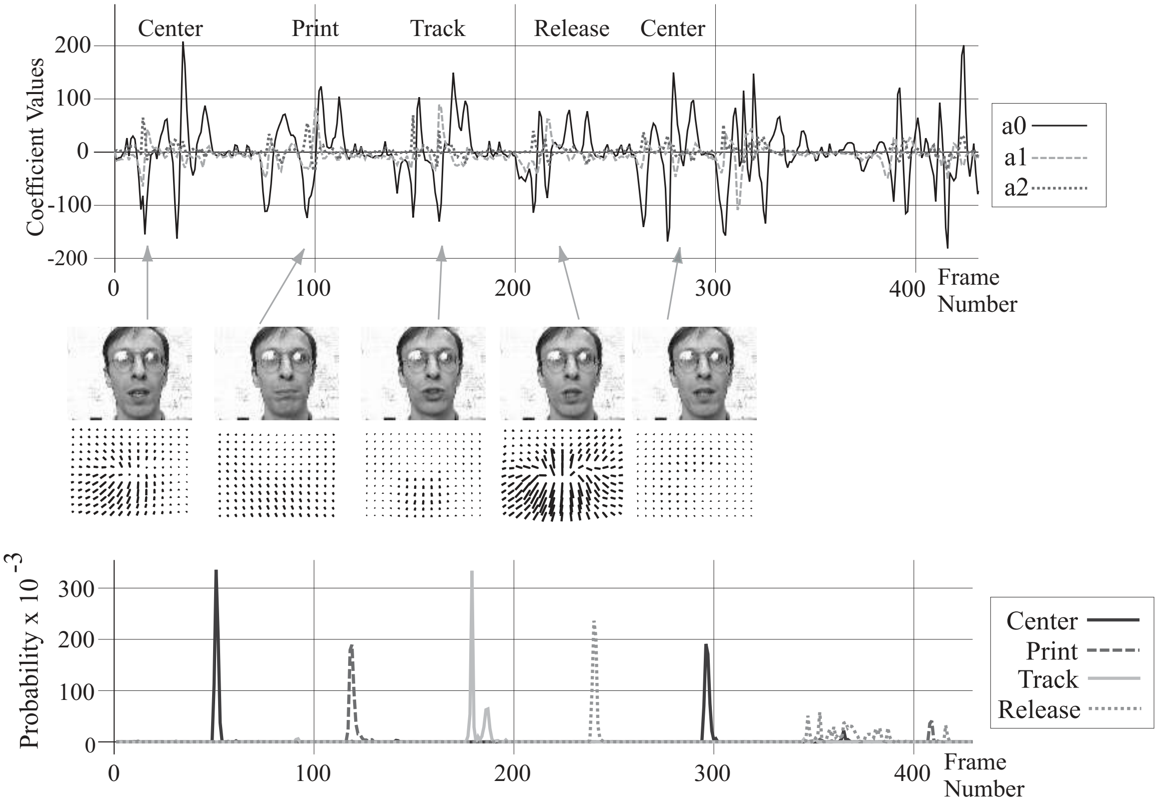
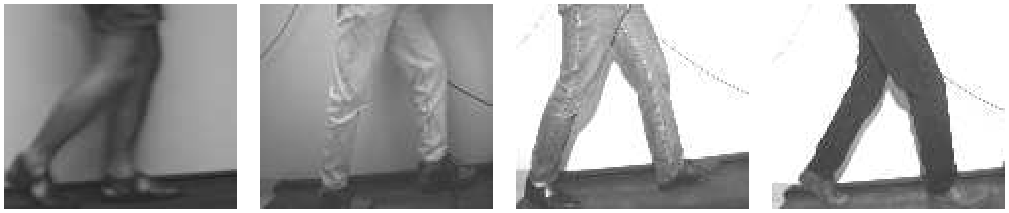


Figure 20: Word detection from mouth motion coefficients: (top) With a test sequence, the model coefficients are estimated between every pair of frames. For the mouth model we used 3 basis flow fields, the coefficients of which are shown for a 400 frame test sequence. (bottom) Recognition probabilities for test words from the mouth motion (see text).

models are constructed by manually aligning the training curves for the same utterance and computing the mean curves for each coefficient (bottom of Figure 19).

Next the basis set was applied to a 400 image test sequence in which the same subject said each of the utterances. Figure 20(top) shows the estimated coefficients. Note that each time instant corresponds to a vector of coefficients that define a flow field for the mouth region.

Recognition is performed using a stochastic curve matching algorithm described in [14]. The method uses the Condensation algorithm [36] to match a set of model curves to an input curve in an on-line fashion while allowing various deformations of the model curves to achieve a match. These deformations, among other things, allow us to take small changes in the rate of motion into account. While Figure 20(top) shows the input coefficient trajectories, Figure 20(bottom) shows the output of the recognition, that is, the probability of each temporal model being recognized. Comparing the “spikes” in this plot with the input trajectories reveals that the method recognizes the completion of each of the test utterances (see [14] for details).



walk-1 *walk-2* *walk-3* *walk-4*

Figure 21: Articulated human motion. The first two images are from the training sequences, and the second two are from test sequences.

Further experimental work with linear parameterized models can be found in work on modeling appearance changes in image sequences [18]. In that work, linear parameterized motion models were used, in conjunction with models for other sources of appearance change in image sequences, to explain image changes between frames in terms of a mixture of causes (see [18] for more details).

5.2.2 Articulated Motion

Like mouths, the articulated motion of human limbs can be large, varied, and difficult to model. Here we construct a view-based model of leg motion and use it to recognize walking motions. We assume that the subject is viewed from the side (though the approach can be extended to cope with other views) and that the image sequence has been stabilized with respect to the torso. Two training and two test sequences (Fig. 21) of subjects walking on a treadmill were acquired with different lighting conditions, viewing position, clothing, and speed of activity. One advantage of motion-based recognition over appearance-based approaches is that it is relatively insensitive to changes such as these.

PCA was performed on a 350-image training set. The first nine basis vectors account for 90% of variance in the training data (see Fig. 22), and are used in our experiments (cf. [4]). A temporal model was constructed for a single walking cycle by manually segmenting and aligning all the cycles present in the training data. This temporal walking model is shown in Figure 23. The motion coefficients were then estimated for the 200 image test sequence. The top two rows of Figure 24 show the images and recovered motions at every 50th frame. Below this is a plot of the first four coefficients over the entire test sequence. The Condensation-based recognition algorithm was used to recognize the walking cycles. The peaks in the bottom plot in Figure 24 correspond to the detection of a completed walking cycle.

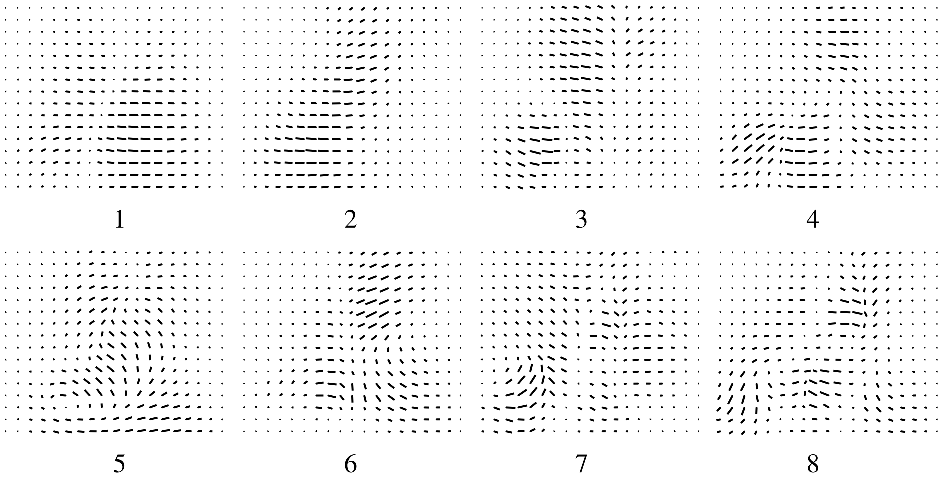


Figure 22: Basis flow fields for the walking sequences.

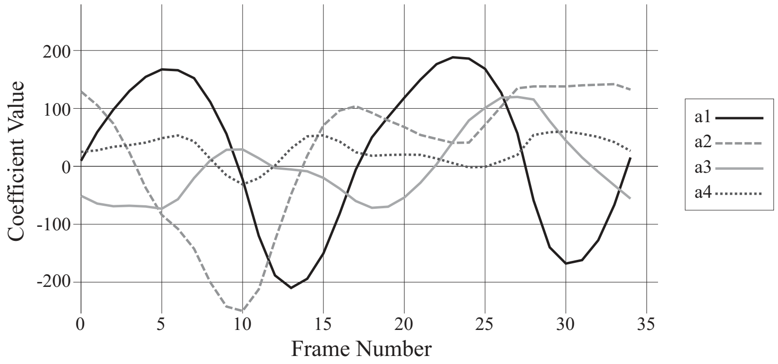


Figure 23: Temporal trajectory model for walking.

6 Conclusion

Linear parameterized motion models can provide effective descriptions of optical flow. They impose strong constraints on the spatial variation of the optical flow within an image region and they provide a concise description of the motion in terms of a small number of linear coefficients. Moreover, the model coefficients can be estimated directly from the image derivatives and do not require the prior computation of dense image motion. The framework described here extends parameterized flow methods to more complex motions.

We have explored the use of linear parameterized models to represent generic, discontinuous motion features including occlusion boundaries and motion bars. The models are applied at every image location in the way that current affine models are employed, and the model coefficients are estimated from the image in exactly the same way as affine motion coefficients are computed, with robust, coarse-to-fine optical flow techniques. Finally, we have shown how to reliably detect the

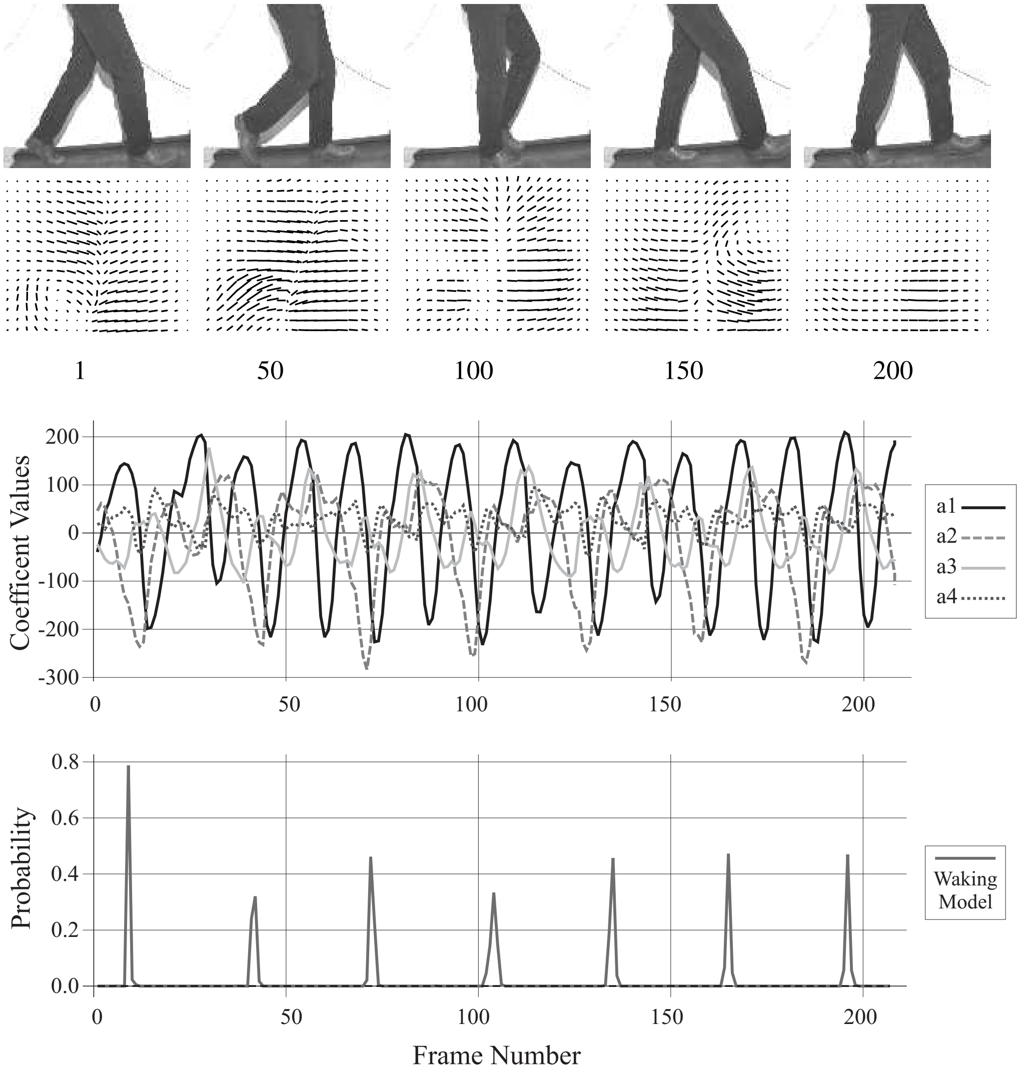


Figure 24: Top two rows: images and estimated flow for every 50th frame in the test sequence. Third row: Plot of the first four motion coefficients for one test sequence. Bottom row: Plot showing the detection of completed walking cycles.

presence of a motion feature from the linear coefficients and how to recover the feature orientation and the relative velocities of the surfaces. This work shows one way to extend regression-based optical flow methods to cope with more complex features and helps bring to light the relationships between static image features and motion features. We have also used this method as a means for initializing a particle filter for detecting and tracking motion boundaries with a higher-dimensional nonlinear model [17].

The framework presented here also extends linear parameterized motion models to object-specific domains (e.g., mouth motion or human walking motions). In these domains, rather than explicitly constructing the model bases, we learn them from sets of training flow fields. Principal component analysis is used to construct low-dimensional representations of the motion classes, the coefficients of which are estimated directly from spatiotemporal image derivatives. Unlike the motion features above, we have applied these models in specific image regions. For example, once a head is tracked and stabilized the motion of the mouth can be estimated using the learned mouth motion model. This model alignment in the image is important, and it may be possible to refine this alignment automatically (see [13]). The temporal behaviour of the model coefficients then provide a rich description of the local motion information. To show this, we have used the time-varying behaviour of the coefficients to construct and recognize various simple temporal "gestures" [14]. These include spoken words with mouth motion, and a walking gait in the context of human locomotion. Speech recognition from lip motion alone is of course a very challenging task and we have demonstrated results in a highly constrained domain. Future work should explore the combination of motion with appearance information and audio.

Future Work

In future work we plan to extend the use of linear parameterized models in several ways. With motion features we plan to improve the detection method with the combined (edge and bar) basis set to use the coefficient behavior in both subspaces to improve detection and parameter estimation. We expect that localization can be greatly improved in this way. Localization can be further improved with the inclusion of information about static image features, because motion edges often coincide with intensity edges.

A related issue concerns the fact that we have only used motion features at a single scale. As one consequence of this, we do not currently detect bars much wider than 8 or 10 pixels, and we do not estimate the width of the bar. It may be desirable to have edge and bar models at a variety of scales that provide better spatial resolution at fine scales and potentially better detection at coarser scales. In this case we would expect that structure could be tracked through scale, where, for example, two edges at one scale would be detected as a bar at a coarser scale. Another way to include multiple scales would be to use a multiscale wavelet basis to represent optical flow [62]. From the wavelet coefficients one could detect motion edges and bars much like intensity edges are detected from image wavelet transforms.

Currently, the coefficients of each image region are estimated independently and it would be interesting to explore the regularization of neighboring coefficients to reduce noise and enforce continuity along contours. However, the exploitation of statistical dependences between events in nearby regions may be best incorporated at a subsequent stage of analysis, like the probabilitic approach described in [17].

With the domain-specific motion models, we assumed that we were given the appropriate image location within which to apply the models. If size and shape of the actual image region is somewhat different from training set, then one might also allow an affine deformation to warp the image data into the subspace (cf. [13]). More generally, given a set of models that characterize a variety of motions in the natural world, we would like to find the appropriate model to use in a given region; this is related to work on object and feature recognition using appearance based models [42].

A number of other research issues remain unanswered. Learned models are particularly useful in situations where optical flow is hard to estimate, but in these situations it is difficult to compute reliable training data. This problem is compounded by the sensitivity of PCA to outliers. PCA also gives more weight to large motions making it difficult to learn compact models of motions with important structure at multiple scales. Future work will explore non-linear models of image motion, robust and incremental learning, and models of motion texture.

Acknowledgements.

We are grateful to Prof. Mubarak Shah for suggesting to us the application of PCA to optical flow fields. DJF was supported by NSERC Canada and by an Alfred P. Sloan Research Fellowship.

References

[1] S. Ayer and H. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In *IEEE International Conference on Computer Vision*, pages 777–784, Boston, MA, 1995. IEEE.

[2] A. Bab-Hadiashar and D. Suter. Robust optical flow computation. *International Journal of Computer Vision*, 29:59–77, 1998.

[3] J.L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1), 1994.

[4] A. Baumberg and D. Hogg. Learning flexible models from image sequences. In J. Eklundh, editor, *European Conf. on Computer Vision, ECCV-94*, volume 800 of *LNCS-Series*, pages 299–308, Stockholm, Sweden, 1994. Springer-Verlag.

[5] S. S. Beauchemin and J. L. Barron. The local frequency structure of 1d occluding image signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (to appear), 1999.

[6] A. J. Bell and T. J. Sejnowski. The independent components of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, December 1997.

[7] J. R. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of Second European Conf. on Comp. Vis.*, pages 237–252. Springer-Verlag, 1992.

[8] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg. A three-frame algorithm for estimating two-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):886–896, September 1992.

[9] D. Beymer. Feature correspondence by interleaving shape and texture computations. In *Proc. Computer Vision and Pattern Recognition, CVPR-96*, pages 921–928, San Francisco, June 1996.

[10] M. Black, F. Bérard, A. Jepson, W. Newman, E. Saund, G. Socher, and M. Taylor. The digital office: Overview. In *AAAI Spring Symposium on Intelligent Environments*, pages 1–6, Stanford, CA, March 1998.

[11] M. J. Black and P. Anandan. Constraints for the early detection of discontinuity from motion. In *Proc. National Conf. on Artificial Intelligence, AAAI-90*, pages 1060–1066, Boston, MA, 1990.

[12] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, January 1996.

[13] M. J. Black and A. D. Jepson. EigenTracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.

[14] M. J. Black and A. D. Jepson. A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In H. Burkhardt and B. Neumann, editors, *European Conf. on Computer Vision, ECCV-98*, volume 1406 of *LNCS-Series*, pages 909–924, Freiburg, Germany, 1998. Springer-Verlag.

[15] M. J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.

[16] M. J. Black, Y. Yacoob, A. D. Jepson, and D. J. Fleet. Learning parameterized models of image motion. In *Proc. Computer Vision and Pattern Recognition, CVPR-97*, pages 561–567, Puerto Rico, June 1997.

[17] M.J. Black and D.J. Fleet. Probabilistic detection and tracking of motion discontinuities. In *Proc. of International Conference on Computer Vision*, pages 551–558, Corfu, Greece, September 1999.

[18] M.J. Black, D.J. Fleet, and Y. Yacoob. Robustly estimating changes in image appearance. *Computer Vision and Image Understanding*, to appear.

[19] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. Computer Vision and Pattern Recognition, CVPR-98*, pages 8–15, Santa Barbara, June 1998.

[20] C. Bregler and S. M. Omohundro. Nonlinear manifold learning for visual speech recognition. In *IEEE International Conference on Computer Vision*, pages 494–499, Boston, MA, 1995. IEEE.

[21] P. J. Burt, J. R. Bergen, R. Hingorani, R. Kolczynski, W. A. Lee, A. Leung, J. Lubin, and H. Shvaytser. Object tracking with a moving camera: An application of dynamic motion analysis. In *Proceedings of the Workshop on Visual Motion*, pages 2–12, Irvine, CA, March 1989.

[22] G. T. Chou. A model of figure-ground segregation from kinetic occlusion. In *IEEE International Conference on Computer Vision*, pages 1050–1057, Boston, MA, 1995. IEEE.

[23] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In H. Burkhardt and B. Neumann, editors, *European Conf. on Computer Vision, ECCV-98*, volume 1406 of *LNCS-Series*, pages 484–498, Freiburg, Germany, 1998. Springer-Verlag.

[24] T. Darrell and A. Pentland. Cooperative robust estimation using layers of support. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):474–487, May 1995.

[25] T. Ezzat and T. Poggio. Facial analysis and synthesis using image-based models. In *International Conference on Automatic Face and Gesture Recognition*, pages 116–121, Killington, Vermont, 1996.

[26] C. L. Fennema and W. B. Thompson. Velocity determination in scenes containing several moving objects. *Computer Vision, Graphics, and Image Processing*, 9:301–315, 1979.

[27] D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5:77–104, 1990.

[28] D. J. Fleet and K. Langley. Computational analysis of non-fourier motion. *Vision Research*, 22:3057–3079, 1994.

[29] W. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:891–906, 1991.

[30] S. Geman and D. E. McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, LII-4:5–21, 1987.

[31] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 1983.

[32] G. Hager and P. Belhumeur. Real-time tracking of image regions with changes in geometry and illumination. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–410, San Francisco, CA, 1996. IEEE.

[33] P. Hallinan. *A deformable model for the recognition of human faces under arbitrary illumination*. PhD thesis, Harvard University, Cambridge, MA, August 1995.

[34] J. G. Harris, C. Koch, E. Staats, and J. Luo. Analog hardware for detecting discontinuities in early vision. *Int. Journal of Comp. Vision*, 4(3):211–223, June 1990.

[35] F. Heitz and P. Bouthemy. Multimodal motion estimation of discontinuous optical flow using markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1217–1232, 1993.

[36] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):2–28, August 1998.

[37] A. Jepson and M. J. Black. Mixture models for optical flow computation. In Ingmer Cox, Pierre Hansen, and Bela Julesz, editors, Partitioning Data Sets: With Applications to Psychology, Vision and Target Tracking, pages 271–286, DIMACS Workshop, April 1993. AMS Pub., Providence, RI.

[38] S. X. Ju, M. J. Black, and A. D. Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 307–314, San Francisco, June 1996.

[39] J. K. Kearney and W. B. Thompson. An error analysis of gradient-based methods for optical flow estimation. *IEEE Pattern Analysis and Machine Intelligence*, 19(2):229–244, 1987.

[40] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, 1981.

[41] C. Nastar, B. Moghaddam, and A. Pentland. Generalized image matching: Statistical learning of physically-based deformations. In B. Buxton and R. Cipolla, editors, *European Conf. on Computer Vision, ECCV-96*, volume 1064 of *LNCS-Series*, pages 589–598, Cambridge, UK, 1996. Springer-Verlag.

[42] S. K. Nayar, S. Baker, and H. Murase. Parametric feature detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 471–477, San Francisco, CA, 1996. IEEE.

[43] R. C. Nelson and R. Polana. Qualitative recognition of motion using temporal texture. *CVGIP: Image Understanding*, 56(1):78–89, 1992.

[44] S. A. Niyogi. Detecting kinetic occlusion. In *IEEE International Conference on Computer Vision*, pages 1044–1049, Boston, MA, 1995. IEEE.

[45] E.P. Ong and M. Spann. Robust optical flow computation based on least-median-of-squares regression. *International Journal of Computer Vision*, 31:51–82, 1999.

[46] P. Perona. Deformable kernels for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:488–499, 1995.

[47] J. L. Potter. Scene segmentation using motion information. *IEEE Trans. S.M.C.*, 5:390–394, 1980.

[48] S. Sclaroff and J. Isidoro. Active blobs. In *Proc. International Conference on Computer Vision*, pages 1146–1153, Mumbai, India, January 1998.

[49] S. Sclaroff and A. P. Pentland. Physically-based combinations of views: Representing rigid and nonrigid motion. In *Proceedings of the Workshop on Motion of Non-rigid and Articulated Objects*, pages 158–164, Austin, Texas, November 1994.

[50] D. Shulman and J.Y. Herve. Regularization of discontinuous flow fields. In *Proceedings of IEEE Workshop on Visual Motion*, pages 81–86, Irvine, CA, March 1989.

[51] A. Spoerri and S. Ullman. The early detection of motion boundaries. In *IEEE International Conference on Computer Vision*, pages 209–218, London, UK, 1987. IEEE.

[52] R. Szeliski and J. Coughlan. Spline-based image registration. *International Journal of Computer Vision*, 22:199–213, 1997.

[53] R. Szeliski and H. Shum. Motion estimation with quadtree splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1199–1211, 1996.

[54] W. B. Thompson, K. M. Mutch, and V. A. Berzins. Dynamic occlusion analysis in optical flow fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:374–383, 1985.

[55] N. Vasconcelos and A. Lippman. A spatiotemporal motion model for video summerization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–366, Santa Barbara, June 1998.

[56] T. Vetter. Learning novel views to a single face image. In *International Conference on Automatic Face and Gesture Recognition*, pages 22–27, Killington, Vermont, 1996.

[57] T. Vetter, M. J. Jones, and T. Poggio. A bootstrapping algorithm for learning linear models of object classes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 40–46, Puerto Rico, June 1997.

[58] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, September 1994.

[59] A. M. Waxman and K. Wohn. Contour evolution, neighbourhood deformation and global image flow: Planar surfaces in motion. *International Journal of Robotics Research*, 4:95–108, 1985.

[60] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 520–526, Puerto Rico, June 1997.

[61] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *Proc. Computer Vision and Pattern Recognition, CVPR-96*, pages 321–326, San Francisco, June 1996.

[62] Y. Wu, T. Kanade, J. Cohn, and C. Li. Optical flow estimation using wavelet motion model. In *Proceedings of the International Conference on Computer Vision*, pages 992–998, Mumbai, India, January 1998.

[63] M. Yamamoto, S. Sato, S. Kuwada, T. Kondo, and Y. Osaki. Incremental tracking of human actions from multiple views. In *Proc. Computer Vision and Pattern Recognition, CVPR-98*, pages 2–7, Santa Barbara, June 1998.