Optical Flow Estimation

David J. Fleet, Yair Weiss

ABSTRACT This chapter provides a tutorial introduction to gradient-based optical flow estimation. We discuss least-squares and robust estimators, iterative coarse-to-fine refinement, different forms of parametric motion models, different conservation assumptions, probabilistic formulations, and robust mixture models.

1 Introduction

Motion is an intrinsic property of the world and an integral part of our visual experience. It is a rich source of information that supports a wide variety of visual tasks, including 3D shape acquisition and oculomotor control, perceptual organization, object recognition and scene understanding [16, 21, 26, 33, 35, 38, 45, 47, 50]. In this chapter we are concerned with general image sequences of 3D scenes in which objects and the camera may be moving. In camera-centered coordinates each point on a 3D surface moves along a 3D path $\vec{X}(t)$. When projected onto the image plane each point produces a 2D path $\vec{x}(t) \equiv (x(t), y(t))^T$, the instantaneous direction of which is the velocity $d\,\vec{x}(t)/dt$. The 2D velocities for all visible surface points is often referred to the *2D motion field* [27]. The goal of *optical flow* estimation is to compute an approximation to the motion field from time-varying image intensity. While several different approaches to motion estimation have been proposed, including correlation or block-matching (e.g, [3]), feature tracking, and energy-based methods (e.g., [1]), this chapter concentrates on gradient-based approaches; see [6] for an overview and comparison of the other common techniques.

2 Basic Gradient-Based Estimation

A common starting point for optical flow estimation is to assume that pixel intensities are translated from one frame to the next,

$$I(\vec{x}, t) \;=\; I(\vec{x} + \vec{u},\, t+1) \;, \tag{1.1}$$

where $I(\vec{x}, t)$ is image intensity as a function of space $\vec{x} = (x, y)^T$ and time t, and $\vec{u} = (u_1, u_2)^T$ is the 2D velocity. Of course, *brightness constancy*

Appears in "Mathematical Models in Computer Vision: The Handbook," N. Paragios, Y. Chen, and O. Faugeras (editors), Chapter 15, Springer, 2005, pp. 239-258

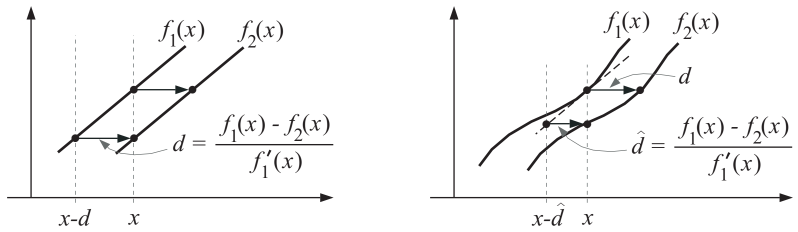
$f_1(x)$
$f_2(x)$
$d = \frac{f_1(x) - f_2(x)}{f_1'(x)}$
x-d
x
$f_1(x)$
$f_2(x)$
d
$\hat{d} = \frac{f_1(x) - f_2(x)}{f_1'(x)}$
x-$\hat{d}$
x

FIGURE 1. The gradient constraint relates the displacement of the signal to its temporal difference and spatial derivatives (slope). For a displacement of a linear signal (left), the difference in signal values at a point divided by the slope gives the displacement. For nonlinear signals (right), the difference divided by the slope gives an approximation to the displacement.

rarely holds exactly. The underlying assumption is that surface radiance remains fixed from one frame to the next. One can fabricate scenes for which this holds; e.g., the scene might be constrained to contain only Lambertian surfaces (no specularities), with a distant point source (so that changing the distance to the light source has no effect), no object rotations, and no secondary illumination (shadows or inter-surface reflection). Although unrealistic, it is remarkable that the brightness constancy assumption (1.1) works so well in practice.

To derive an estimator for 2D velocity $\vec{u}$, we first consider the 1D case. Let $f_1(x)$ and $f_2(x)$ be 1D signals (images) at two time instants. As depicted in Fig. 1, suppose further that $f_2(x)$ is a translated version of $f_1(x)$; i.e., let $f_2(x) = f_1(x-d)$ where d denotes the translation. A Taylor series expansion of $f_1(x-d)$ about x is given by

$$f_1(x-d) \;=\; f_1(x) - d\, f_1'(x) + O(d^2 f_1'') \;, \tag{1.2}$$

where $f' \equiv d\,f(x)/dx$. With this expansion we can rewrite the difference between the two signals at location x as

$$f_1(x) - f_2(x) \;=\; d\, f_1'(x) + O(d^2 f_1'') \;.$$

Ignoring second- and higher-order terms, we obtain an approximation to d:

$$\hat{d} \;=\; \frac{f_1(x) - f_2(x)}{f_1'(x)} \;. \tag{1.3}$$

The 1D case generalizes straightforwardly to 2D. As above, assume that the displaced image is well approximated by a first-order Taylor series:

$$I(\vec{x}+\vec{u}, t+1) \;\approx\; I(\vec{x}, t) + \vec{u}\cdot\nabla I(\vec{x}, t) + I_t(\vec{x}, t) \;, \tag{1.4}$$

where $\nabla I \equiv (I_x, I_y)$ and I_t denote spatial and temporal partial derivatives of the image I, and $\vec{u} = (u_1, u_2)^T$ denotes the 2D velocity. Ignoring higher-order terms in the Taylor series. and then substituting the linear approximation into (1.1), we obtain [28]

$$\nabla I(\vec{x},t)\cdot\vec{u} + I_t(\vec{x},t) \;=\; 0\ . \tag{1.5}$$

Equation (1.5) relates the velocity to the space-time image derivatives at one image location, and is often called the *gradient constraint equation*. If one has access to only two frames, or cannot estimate I_t, it is straightforward to derive a closely related gradient constraint, in which $I_t(\vec{x},t)$ in (1.5) is replaced by $\delta I(\vec{x},t) \equiv I(\vec{x},t+1) - I(\vec{x},t)$ [34].

Intensity Conservation

Tracking points of constant brightness can also be viewed as the estimation of 2D paths $\vec{x}(t)$ along which intensity is conserved:

$$I(\vec{x}(t),t) \;=\; c\ , \tag{1.6}$$

the temporal derivative of which yields

$$\frac{d}{d\,t}\,I(\vec{x}(t),t) \;=\; 0\ . \tag{1.7}$$

Expanding the left-hand-side of (1.7) using the chain rule gives us

$$\frac{d}{d\,t}\,I(\vec{x}(t),t) \;=\; \frac{\partial I}{\partial x}\frac{d\,x}{d\,t} + \frac{\partial I}{\partial y}\frac{d\,y}{d\,t} + \frac{\partial I}{\partial t}\frac{d\,t}{d\,t} \;=\; \nabla I\cdot\vec{u} + I_t\ , \tag{1.8}$$

where the path derivative is just the optical flow $\vec{u} \equiv (dx/dt, dy/dt)^T$. If we combine (1.7) and (1.8) we obtain the gradient constraint equation (1.5).

Least-Squares Estimation

Of course, one cannot recover $\vec{u}$ from one gradient constraint since (1.5) is one equation with two unknowns, u_1 and u_2. The intensity gradient constrains the flow to a one parameter family of velocities along a line in *velocity space*. One can see from (1.5) that this line is perpendicular to ∇I, and its perpendicular distance from the origin is $|I_t|/||\nabla I||$.

One common way to further constrain $\vec{u}$ is to use gradient constraints from nearby pixels, assuming they share the same 2D velocity. With many constraints there may be no velocity that simultaneously satisfies them all, so instead we find the velocity that minimizes the constraint errors. The least-squares (LS) estimator minimizes the squared errors [34]:

$$\boxed{E(\vec{u}) = \sum_{\vec{x}} g(\vec{x})\,[\vec{u}\cdot\nabla I(\vec{x},t) + I_t(\vec{x},t)]^2\ ,} \tag{1.9}$$

where $g(\vec{x})$ is a weighting function that determines the *support* of the estimator (the region within which we combine constraints). It is common to let $g(\vec{x})$ be Gaussian in order to weight constraints in the center of the neighborhood more highly, giving them more influence. The 2D velocity $\hat{u}$ that minimizes $E(\vec{u})$ is the least squares flow estimate.

The minimum of $E(\vec{u})$ can be found from its critical points, where its derivatives with respect to $\vec{u}$ are zero; i.e.,

$$\begin{aligned} \frac{\partial E(u_1, u_2)}{\partial u_1} &= \sum_{\vec{x}} g(\vec{x}) \left[u_1 {I_x}^2 + u_2 I_x I_y + I_x I_t\right] &= 0 \\ \frac{\partial E(u_1, u_2)}{\partial u_2} &= \sum_{\vec{x}} g(\vec{x}) \left[u_2 {I_y}^2 + u_1 I_x I_y + I_y I_t\right] &= 0\,. \end{aligned}$$

These equations may be rewritten in matrix form:

$$\mathbf{M}\,\vec{u} \;=\; \vec{b}\,, \tag{1.10}$$

where the elements of $\mathbf{M}$ and $\vec{b}$ are:

$$\mathbf{M} \;=\; \begin{bmatrix} \sum g\, {I_x}^2 & \sum g\, I_x I_y \\ \sum g\, I_x I_y & \sum g\, {I_y}^2 \end{bmatrix}\,, \qquad \vec{b} \;=\; -\begin{pmatrix} \sum g\, I_x I_t \\ \sum g\, I_y I_t \end{pmatrix}\,.$$

When $\mathbf{M}$ has rank 2, then the LS estimate is $\hat{u} = \mathbf{M}^{-1}\vec{b}$.

Implementation Issues

Usually we wish to estimate optical flow at every pixel, so we should express $\mathbf{M}$ and $\vec{b}$ as functions of position $\vec{x}$, i.e., $\mathbf{M}(\vec{x})\,\vec{u}(\vec{x}) = \vec{b}(\vec{x})$. Note that the elements of $\mathbf{M}$ and $\vec{b}$ are local sums of products of image derivatives. An effective way to estimate the flow field is to first compute derivative images through convolution with suitable filters. Then, compute their products (${I_x}^2$, I_xI_y, ${I_y}^2$, I_xI_t and I_yI_t), as required by (1.10). These quadratic images are then convolved with $g(\vec{x},)$ to obtain the elements of $\mathbf{M}(\vec{x})$ and $\vec{b}(\vec{x})$.

In practice, the image derivatives will be approximated using numerical differentiation. It is important to use a consistent approximation scheme for all three directions [13]. For example, using simple forward differencing (i.e., $\hat{I}_x = I(x,y) - I(x+1,y)$) will not give a consistent approximation as the x, y and t derivatives will be centered at different locations in the xyt-cube [27]. Another practicality worth mentioning is that some image smoothing is generally useful prior to numerical differentiation (and can be incorporated into the derivative filters). This can be justified from the first-order Taylor series approximation used to derive (1.5). By smoothing the signal, one hopes to reduce the amplitudes of higher-order terms in the image and to avoid some related problems with temporal aliasing.

Aperture Problem

When $\mathbf{M}$ in (1.10) is rank deficient one cannot solve for $\vec{u}$. This is often called the aperture problem as it invariably occurs when the support $g(x)$ is

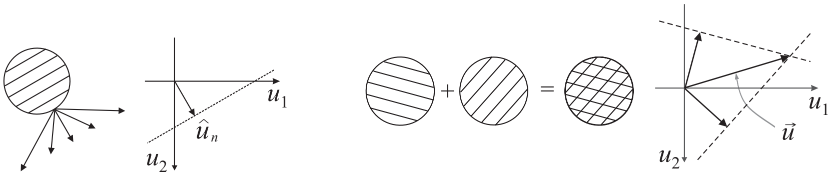

FIGURE 2. (left) A single moving grating viewed through a circular aperture is consistent with all 2D velocities along a line in velocity space. (right) With two drifting gratings there are multiple constraint lines that intersect to uniquely constrain the 2D velocity. (After [2])

sufficiently local. However, the important issue is not the width of support, but rather the dimensionality of the image structure. Even for large regions, if the image is one-dimensional then **M** will be singular. As depicted in Fig. 2 (left); when each image gradient within a region has the same spatial direction, it is easy to see that $rank[\mathbf{M}] = 1$. Moreover, note that a single gradient constraint only provides the normal component of $\vec{u}$,

$$\vec{u}_n \;=\; \frac{-I_t}{||\nabla I||}\,\frac{\nabla I}{||\nabla I||} \;.$$

When there exist constraints with two or more gradient directions, as depicted in Fig. 2 (right), then the different constraint lines intersect to uniquely constrain the 2D velocity.

3 Iterative Optical Flow Estimation

Equation (1.9) provides an optimal solution, but not to our original problem. Remember that we ignored high-order terms in the derivation of (1.3) and (1.5). As depicted in Fig. 1, if f_1 is linear then $d = \hat{d}$. Otherwise, to leading order, the accuracy of the estimate is bounded by the magnitude of the displacement and the second derivative of f_1:

$$|\hat{d} - d| \;\leq\; \frac{d^2\,|f_1''(x)|}{2\,|f_1'(x)|} \;+\; O(d^3)\;. \tag{1.11}$$

For a sufficiently small displacement, and bounded $|f_1''/f_1'|$, we expect reasonably accurate estimates. This suggests a form of Gauss-Newton optimization in which we use the current estimate to *undo* the motion, and then we reapply the estimator to the *warped* signals to find the residual motion. This continues until the residual motion is sufficiently small.

In 2D, given an estimate of the optical flow field $\vec{u}^{\,0}$, we create a *warped* image sequence $I^0(\vec{x}, t)$:

$$I^0(\vec{x},\, t+\delta t) \;=\; I(\vec{x} + \vec{u}^{\,0}\delta t, t+\delta t)\;, \tag{1.12}$$

where δt is the time between consecutive frames. (In practice, we only need to warp enough frames for temporal differentiation.) Assuming that $\vec{u} = \vec{u}^0 + \delta\vec{u}$, it is straightforward to see from (1.1) and (1.12) that

$$I^0(\vec{x}, t) \;=\; I^0(\vec{x} + \delta\vec{u}, t+1)\ . \tag{1.13}$$

If $\delta\vec{u} = \vec{0}$, then clearly I^0 would be constant through time (assuming brightness constancy). Otherwise, we can estimate the residual flow using

$$\delta\hat{u} = \mathbf{M}^{-1}\,\vec{b} \tag{1.14}$$

where $\mathbf{M}$ and $\vec{b}$ are computed by taking spatial and temporal derivatives (differences) of I^0. The refined optical flow estimate then becomes

$$\vec{u}^{\,1} \;=\; \vec{u}^{\,0} + \delta\hat{u}\ .$$

In an iterative manner, this new flow estimate is then used to rewarp the original sequence (as in (1.12)), and another residual flow can be estimated.

This iteration yields a sequence of approximate objective functions that converge to the desired objective function [10]. At iteration j, given the estimate $\vec{u}^{\,j}$ and the warped sequence I^j, our desired objective function is

$$\begin{aligned} E(\delta\vec{u}) \;&=\; \sum_{\vec{x}} g(\vec{x})\ \left[I(\vec{x},t) - I(\vec{x}+\vec{u}^{\,j}+\delta\vec{u},\, t+1)\right]^2 \qquad (1.15)\\ &=\; \sum_{\vec{x}} g(\vec{x})\ \left[I^j(\vec{x},t) - I^j(\vec{x}+\delta\vec{u},\, t+1)\right]^2 \\ &\approx\; \sum_{\vec{x}} g(\vec{x})\ \left[\nabla I^j(\vec{x},t)\cdot\delta\vec{u} + I^j_t(\vec{x},t)\right]^2 \;\;\equiv\;\; \tilde{E}(\delta\vec{u})\ . \qquad (1.16)\end{aligned}$$

The gradient approximation to the difference in (1.15) gives an approximate objective function $\tilde{E}$. From (1.11) one can show that $\tilde{E}$ approximates E to second-order in the magnitude of the residual flow, $\delta\vec{u}$. The approximation error vanishes as $\delta\vec{u}$ is reduced to zero. The iterative refinement with rewarping reduces the residual motion at each iteration so that the approximate objective function converges to the desired objective function, and hence the flow estimate converges to the optimal LS estimate (1.15).

The most expensive step at each iteration is the computation of image gradients and the matrix inverse in (1.14). One can, however, formulate the problem so that the spatial image derivatives used to form $\mathbf{M}$ are taken at time t, and as such, do not depend on the current flow estimate $\vec{u}^{\,j}$ [23]. To see this, note that the spatial deriatives are computed at time t and it is straightforward to see that $I(\vec{x},t) = I^j(\vec{x},t)$. Of course $\vec{b}$ in (1.14) will always depend on the warped image sequence and must be recomputed at each iteration. In practice, when $\mathbf{M}$ is not recomputed from the warped sequence then the spatial and temporal derivatives will not centered at the same location in (x, y, t) and hence more iterations may be needed.

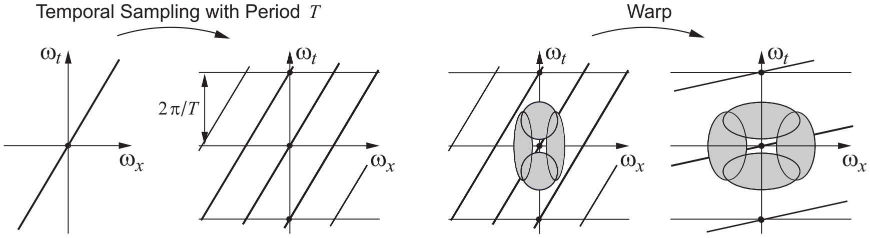

FIGURE 3. (Left) The spectrum of a translating signal is nonzero on a line in the frequency domain. Temporal sampling introduces spectral replicas, causing aliasing for higher speeds (steeper slopes). (Right) The problem may be avoided by blurring the images before computing derivatives. The spectra of such *coarse-scale* filters will be insensitive to the replicas. Velocity estimates from the coarse scale are used to warp the images, thereby undoing much of the motion. Finer-scale derivative filters can now be used to estimate the residual motion. (After [43])

Temporal Aliasing and Coarse-To-Fine Refinement

In practice, our images have temporal sampling rates lower than required by the sampling theorem to uniquely reconstruct the continuous signal. As a consequence, temporal aliasing is a common problem in motion estimation.

The spectrum of a translating signal is confined to a plane through the origin in the frequency domain [15, 51]. That is, if we construct a space-time signal $f(\vec{x},t)$ by translating a 2D signal $f_0(\vec{x})$ with velocity $\vec{u}$, i.e., $f(\vec{x},t) = f_0(\vec{x} - \vec{u}t)$, one can show that the space-time Fourier transform of $f(\vec{x},t)$ is given by

$$F(\omega_x, \omega_y, \omega_t) \;=\; F_0(\omega_x, \omega_y)\,\delta(u_1\omega_x + u_2\omega_y + \omega_t)\ , \tag{1.17}$$

where F_0 is the 2D Fourier transform of f_0 and $\delta()$ is a Dirac delta. Equation (1.17) shows that the spectrum is nonzero only on a plane, the orientation of which gives the velocity. When the continuous signal is sampled in time, replicas of the spectrum are introduced at intervals of $2\pi/T$ radians, where T is the time between frames (see Fig. 3 (left)). It is easy to see how this causes problems; i.e., the derivative filters may be more sensitive to the spectral replicas at high spatial frequencies than to the original spectrum on the plane through the origin.

This suggests a simple approach to aliasing problems [3, 7]. Optical flow can be estimated at the coarsest scale of a Gaussian pyramid, where the image is significantly blurred, and the velocity is much slower (due to subsampling). The coarse-scale estimate can be used to warp the next (finer) pyramid level to *stabilize* its motion. Since the velocities after warping are slower, as shown in Fig. 3 (right)), a wider low-pass frequency band will be free of aliasing. One can therefore use derivatives at the finer scale to estimate the residual motion. This coarse-to-fine estimation continues until the finest level of the pyramid (the original image) is reached. Mathematically, this is identical to iterative refinement except that each scale's estimate must be up-sampled and interpolated before warping the next finer scale.

While widely used, coarse-to-fine methods have their drawbacks, usually stemming from the fact that fine-scale estimates can only be as reliable as their coarse-scale precursors; a poor estimate at one scale provides a poor initial guess at the next finer scale, and so on. That said, when aliasing does occur, one must use some mechanism such as coarse-to-fine estimation to avoid local minima in the optimization.

4 Robust Motion Estimation

The LS estimator is optimal when the gradient constraint errors, i.e.,

$$e(\vec{x}) \;\equiv\; \vec{u} \cdot \nabla I(\vec{x}, t) + I_t(\vec{x}, t) \;, \tag{1.18}$$

are mean-zero Gaussian, and the errors in different constraints are independent and identically distributed (IID). Not surprisingly, this is a fragile assumption. For example, brightness constancy is often violated due to changing surface orientation, specularities reflections, or time-varying shadows. When there is significant depth variation in the scene, the constant motion model will be extremely poor, especially at occlusion boundaries.

LS estimators are not suitable when the distribution of gradient constraint errors is heavy-tailed, as they are sensitive to small numbers of measurement outliers [24, 32]. It is therefore often crucial that the quadratic estimator in (1.9) be replaced by a robust estimator, $\rho(\cdot)$, which limits the influence of constraints with larger errors (e.g., see [5, 9, 41]):

$$E(\vec{u}) \;=\; \sum_{x,y} g(\vec{x}) \; \rho(e(\vec{x}),\, \sigma) \;. \tag{1.19}$$

For example, Black and Anandan [9] used the redescending Geman-McClure estimator [20], $\rho(e, \sigma) \;=\; e^2/(e^2 + \sigma^2)$, where σ^2 determines the range of constraint errors for which influence is reduced.

Among the various ways one might minimize (1.19), one very useful approach takes the form of iteratively reweighted least-squares [32]. In short, this is an iterative solution in which the weights $g(\vec{x})$ in (1.9) are scaled by a weight function that downweights those constraints that are inconsistent (i.e., have large errors) with the current motion estimate. Often it is also useful to anneal the optimization, wherein σ^2 starts large, and is then slowly decreased to achieve greater robustness.

5 Motion Models

Thus far we have assumed that the 2D velocity is constant in local neighbourhoods. Nevertheless, even for small regions this is often a poor assumption. We now consider generalizations to more interesting motion models.

Affine Model

General first-order affine motion is usually a better model of local motion than a translational model (e.g., [7, 9, 17]). An affine velocity field centered at location $\vec{x}_0$ can be expressed in matrix form as

$$\vec{u}(\vec{x};\vec{x}_0) \;=\; \mathbf{A}(\vec{x};\vec{x}_0)\;\vec{c}\;, \tag{1.20}$$

where $\vec{c} = (c_1, c_2, c_3, c_4, c_5, c_6)^T$ are the motion model parameters, and

$$\mathbf{A}(\vec{x};\,\vec{x}_0) \;=\; \begin{bmatrix} 1 & 0 & x-x_0 & y-y_0 & 0 & 0 \\ 0 & 1 & 0 & 0 & x-x_0 & y-y_0 \end{bmatrix} .$$

Combining (1.20) and (1.5) yields the gradient constraint equation

$$\nabla I(\vec{x},t)\,\mathbf{A}(\vec{x};\vec{x}_0)\;\vec{c} + I_t(\vec{x},t) \;=\; 0\;,$$

for which the LS estimate for the neighbourhood has the form

$$\hat{c} \;=\; \mathbf{M}^{-1}\,\vec{b}\;, \tag{1.21}$$

where now $\mathbf{M}$ and $\vec{b}$ are given by

$$\mathbf{M} \;=\; \sum_{\vec{x}} g\,\mathbf{A}^T\nabla I^T\,\nabla I\mathbf{A} \quad , \quad \vec{b} \;=\; -\sum_{\vec{x}} g\,\mathbf{A}^T\nabla I^T\,I_t\;.$$

When $\mathbf{M}$ is rank deficient there is insufficient image structure to estimate the six unknowns. Affine models often require larger support than constant models, and one may need a robust estimator instead of the LS estimator.

Iterative refinement is also straightforward with affine motion models. Let the optimal affine motion be $\vec{u} \;=\; \mathbf{A}\,\vec{c}$, and let the affine estimate at iteration j be $\vec{u}^{\,j} \;=\; \mathbf{A}\,\vec{c}^{\,j}$. Because the flow is linear in the motion parameters, it follows that $\delta\vec{u} \equiv \vec{u} - \vec{u}^{\,j}$ and $\delta\vec{c} \equiv \vec{c} - \vec{c}^{\,j}$ satisfy

$$\delta\vec{u} \;=\; \mathbf{A}\delta\vec{c}\;. \tag{1.22}$$

Accordingly, defining $I^{\,j}(\vec{x},t)$ to be the original sequence $I(\vec{x},t)$ warped by $\vec{u}^{\,j}$ as in (1.12) we use the same LS estimator as in (1.21), but with I and $\hat{c}$ replaced by I^j and $\delta\hat{c}$. The updated LS estimate is then $\vec{c}^{\,j+1} = \vec{c}^{\,j} + \delta\hat{c}$.

Low-Order Parametric Deformations

There are many other polynomial and rational deformations that make useful motion models. *Similarity deformations*, comprising translation (d_1, d_2), 2D rotation θ, and uniform scaling by s are a special case of the affine model,

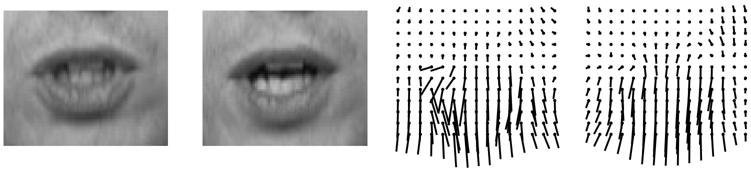

FIGURE 4. (a,b) Mouth regions of two consecutive images of a person speaking. (c) Flow field estimated using dense optical flow method. (d) Flow field estimated using the learned model with 6 basis flow fields. (After [16])

but still very useful in practice. In a neighbourhood centred at $\vec{x}_0$ it has the same form as (1.20), but with $\vec{c} = (d_1,\, d_2,\, s\,\cos\theta,\, s\,\sin\theta)^T$ and

$$\mathbf{A}(\vec{x};\,\vec{x}_0) \;=\; \begin{bmatrix} 1 & 0 & x-x_0 & -y+y_0 \\ 0 & 1 & y-y_0 & x-x_0 \end{bmatrix} .$$

With this linear form, one can solve directly for $\vec{c}$ using linear least-squares, and then compute the similarity parameters d_1, d_2, s, and θ.

Another useful motion model is the *projective deformation* (or homography) [7], which captures image deformations of a 3D plane under camera rotation and translation. See [**?**] for a discussion of homographies and related motion models.

Learned Subspace Models

Many objects exhibit complex motions that are not well modeled by smooth polynomials. For example Fig. 4(a,b) shows two frames of a mouth during speech, for which non-rigidity, occlusion, and fast speeds make flow estimation difficult. Interestingly, the regression framework above extends to diverse types of complex 2D motions with the use of basis flow fields, $\{\vec{b}_j(\vec{x})\}_{j=1}^J$, such that the local optical flow field is expressed as

$$\vec{u}(\vec{x}) \;=\; \sum_{j=1}^{J} c_j\,\vec{b}_j(\vec{x}) \;. \tag{1.23}$$

In this context, optical flow estimation reduces to the estimation of the linear coefficients $\vec{c}$, analogous to the affine model discussed above.

In [16] a motion basis was learned for human mouths. This was accomplished by applying a robust estimator with a generic smoothness model [9] to mouths to obtain training data (e.g., see Fig. 4(c)). The principal components of the ensemble of training flow fields were then extracted and used as the basis. Figure 4(d) shows the optical flow obtained with the subspace model and a robust estimator. The model was found to greatly increase the quality of the optical flow estimates, and the temporal variation in the subspace coefficients were then used to recognize linguistic events.

General Differentiable Warps

In general, one can formulate area-based regression in terms of warp functions $w(\vec{x};\vec{p})$ that are not necessarily smooth in space, nor linear in the warp parameters $\vec{p}$. One can parametrize the warp as a function of time, or assume the two-frame case:

$$I(\vec{x},t) \;=\; I(w(\vec{x};\vec{p}),\, t+1)\ . \tag{1.24}$$

The warp functions must be differentiable with respect to $\vec{p}$. To develop an efficient estimation algorithm, one may need to further constrain w to be invertible (e.g., see [23]).

6 Global Smoothing

While area-based regression is commonly used, some of the earliest formulations of optical flow estimation assumed smoothness through nonparametric motion models, rather than an explicit parametric model in each local neighbourhood (e.g., see [27, 36, 42]). One such energy functional was proposed by Horn and Schunck [28]:

$$E(\vec{u}) \;=\; \int (\nabla I \cdot \vec{u} + I_t)^2 \;+\; \lambda \left(||\nabla u_1||^2 + ||\nabla u_2||^2 \right)\; dx\, dy\ . \tag{1.25}$$

A key advantage of global smoothing is that it enables propagation of information over large distances in the image. In image regions of nearly uniform intensity, such as a blank wall or tabletop, local methods will often yield singular (or poorly conditioned) systems of equations. Global methods can *fill in* the optical flow from nearby gradient constraints.

Equation (1.25) can be minimized directly with discrete approximations to the integral and the derivatives in (1.25). Thie yields a large system of linear equations that may be solved through iterative methods such as *Gauss-Seidel* or *SOR overrelaxation* [22]. Alternatively one can solve the corresponding Euler-Lagrange (PDE) equations under reflecting boundary conditions (e.g., [11, 42]). Recent extensions to global methods include robust penalty functions (for data and smoothness terms), the use of coarse-to-fine search for optimization, and the incorporation of stronger local constraints on the motion, resulting in impressive optical flow estimates [11].

The main disadvantage of global methods is computational efficiency. Even with more efficient optimization algorithms (e.g. [46, 53]) the computational cost is far higher than with local methods. Whether this is justified may depend on the image domain and the need for dense optical flow. Another problem is in the setting of the *regularization parameter* λ that determines the amount of desired smoothing (similar problems arise in choosing the support width for area-based regression). Prior knowledge on the smoothness of flow can be useful here, and more sophisticated methods might be used to estimate (or marginalize) the regularization parameter.

7 Conservation Assumptions

All of the above formulations assumed intensity conservation. Nevertheless, gradient constraints may be used to track any differentiable image property.

Higher-Order Derivative Constraints

Some techniques assume that image gradients are conserved (e.g., [36, 43, 48]). This two further constraints at each pixel, i.e.,

$$\begin{aligned} u_1 I_{xx} + u_2 I_{xy} + I_{xt} &= 0 \\ u_1 I_{xy} + u_2 I_{yy} + I_{yt} &= 0 \; . \end{aligned} \tag{1.26}$$

These are useful insofar as they provide more constraints with which to estimate motion parameters. Conversely, higher-order derivatives are often extremely noisy, and the conservation of ∇I implies that the motion field has no first-order deformation (e.g., rotation). Intensity conservation (1.7), by comparison, assumes only that the image motion is smooth.

Phase-Based Methods

Phase-based methods [17, 18] are based on an initial decomposition of the image into band-pass channels, like those produced by quadrature-pair filters in steerable pyramids [19]. While multi-scale representations are commonly used for flow estimation, a further decomposition into orientation bands, yields more local constraints, often with better signal-to-noise ratios. Complex-valued band-pass images can be represented as real and imaginary images, or in terms of amplitude and phase images. Figure 5 shows the real-part of a 1D band-pass signal, along with its amplitude and phase. Amplitude encodes the magnitude of local signal modulation, while phase encodes the local structure of the signal (e.g., zero-crossings, peaks, etc).

Phase-based methods assume conservation of phase in each band-pass channel. The phase-based gradient constraint, given a complex-valued band-pass channel, $r(\vec{x}, t)$, with phase $\phi(\vec{x}, t) \equiv \arg[r(\vec{x}, t)]$, is simply

$$\nabla\phi(\vec{x}, t) \cdot \vec{u} + \phi_t(\vec{x}, t) \;=\; 0 \; . \tag{1.27}$$

These may be combined to estimate optical flow using any of the estimators above. In practice, because phase is a multi-function, only uniquely defined on intervals of width 2π, explicit differentiation is difficult. Instead, it is convenient to exploit the following identities for computing spatial derivatives and temporal differences,

$$\frac{\partial\phi(\vec{x}, t)}{\partial x} \;=\; \frac{\mathrm{Im}[r_x(\vec{x}, t)\, r^*(\vec{x})]}{|r(\vec{x})|^2} \quad , \qquad \delta\phi(\vec{x}, t) \;=\; \arg[r(\vec{x}, t+1)\, r^*(\vec{x}, t)] \; .$$

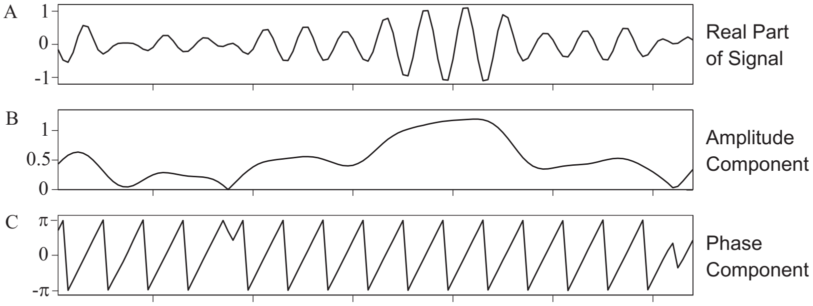

FIGURE 5. A band-pass filtered 1D signal can be expressed using its amplitude and phase signals. Note the linearity of phase over large spatial extents.

where Im$[r]$ denotes the imaginary part of r, r^* is the complex-conjugate of r, and $r_x \equiv \partial r/\partial x$. Compared to phase, $r(\vec{x}, t)$ is relatively easy to differentiate and interpolate [15, 17].

Phase has a number of appealing properties for optical flow estimation. First, phase is amplitude invariant, and therefore quite stable when significant changes in contrast and mean intensity occur between frames. Second, phase is approximately linear over relatively large spatial extents, and has very few critical points where the gradient is zero. This is important as it implies that more gradient constraints may be available, and that the range of velocities that can be estimated is significantly larger than with image derivatives. This also improves the accuracy of gradient-based estimates, reducing the number of iterations required for refinement. Phase has also been shown to be stable with respect to first-order deformations of the image from one time to the next [18]. Both the expected spatial extent of phase linearity and the stability of phase are determined, in part, by filter bandwidth. The main disadvantages of phase concern the computational expense of the band-pass filters, and the spatial support of the filters near occlusion boundaries and fine-scale objects.

Brightness Variations

While contrast normalization, or the use of phase, provides some degree of invariance with respect to deviations from brightness constancy, more significant variations in brightness must be modeled explicitly. The models may be object specific, to model objects under different lighting conditions [23], poses or configurations [10]. Alternatively, the models may be physics-based [25], or they may be generic models for smooth mean and contrast variations [37]. Despite the wide-spread use of brightness constancy these models may be extremely useful for certain domains.

8 Probabilistic Formulations

One problem with the above estimators is that, although they provide useful estimates of optical flow, they do not provide confidence bounds. Nor do they show how to incorporate any prior information one might have about motion to further constrain the estimates. As a result, one may not be able to propagate flow estimates from one time to the next, nor know how to weight them when combining flow estimates from different information sources. These issues can be addressed with a probabilistic formulation.

The cost function (1.16) has a simple probabilistic interpretation. Up to normalization constants, it corresponds to the log likelihood of a velocity under the assumption that intensity is conserved up to Gaussian noise.

$$I(\vec{x},t) \;=\; I(\vec{x}+\vec{u},t+1)+\eta \;. \tag{1.28}$$

If we assume that the same velocity $\vec{u}$ is shared by all pixels within a neighbourhood, that η is white Gaussian noise with standard deviation σ, and uncorrelated at different pixels, we obtain the conditional density

$$p(I \mid \vec{u}) \;\propto\; e^{-\frac{1}{2\sigma^2}E(\vec{u})} \;, \tag{1.29}$$

where $E(\vec{u})$ is the LS objective function (1.16). To obtain further insight into this likelihood function, we again approximate E to second order using $\tilde{E}$ as in (1.15). Under this approximation the likelihood function is Gaussian with mean $\mathbf{M}^{-1}\,\vec{b}$ and covariance matrix $\mathbf{M}^{-1}$.

The approximate covariance matrix $\mathbf{M}^{-1}$ defines an uncertainty ellipse around the estimated optical flow. These uncertainties can be propagated to subsequent frames, or to other spatial scales [44]. They can also be used directly in algorithms for 3D reconstruction [29]. (See [55] for a more detailed discussion of likelihood functions for probabilistic optical flow estimation.)

The probabilistic formulation also allows one to introduce *prior information.* Equation (1.29) can be combined with a prior probability distribution over local velocities. For example, a very useful prior model is that the local flow tends to be *slow* (e.g. [44]). This is convenient to model with a zero-mean Gaussian distribution,

$$p(\vec{u}) \;\propto\; e^{\frac{1}{2\sigma_p^2} - \|\vec{u}\|^2} \;. \tag{1.30}$$

Combining this prior probability with the approximate likelihood function (1.29) gives us a Gaussian posterior probability whose mean (and mode) is

$$\vec{u} \;=\; (\mathbf{M}+\lambda I)^{-1}\,\vec{b} \;, \tag{1.31}$$

where λ is the ratio of the noise and prior variances, $\lambda = \sigma^2/\sigma_p^2$. Note that this Bayesian estimate will actually be biased, and will not correctly estimate the speed or direction of patterns where the local uncertainty is large. This has the benefit that it dampens the estimates to help avoid divergence in iterative refinement and tracking. Interestingly, many "illusions" in human motion perception can actually be explained with a prior favoring slow motions and a Bayesian model of inference [56].

Total Least-Squares

When one assumes significant image noise that contaminates spatial as well as temporal derivatives, then the maximum likelihood motion estimate given a collection of space-time image gradients is given by *total-least-squares* (TLS) [40, 52]. If we view velocity as a unit direction in space-time, or in 3D homogeneous coordinates $\vec{v} \equiv \alpha(u_1, u_2, 1)$, $\alpha \in \mathcal{R}$, and denote the space-time image gradient $\vec{o}_k \equiv (\nabla I(\vec{x}_k, t), I_t(\vec{x}_k, t))^T$, then the gradient constraint becomes $\vec{o}_k^T \vec{v} = 0$. The sum or squared constraint errors is then

$$E(\vec{v}) \;=\; \vec{v}^T \mathbf{S}\, \vec{v} \quad , \qquad \text{where} \quad \mathbf{S} = \sum_k \vec{o}_k\, \vec{o}_k^{\,T} \; . \tag{1.32}$$

The TLS solution is obtained by minimizing $E(\vec{v})$ in (1.32), subject to the constraint $||\vec{v}|| = 1$ to avoid the trivial solution. The solution is given by the eigenvector corresponding to the minimum eigenvalue of $\mathbf{S}$. This approach has been called tensor-based, with $\mathbf{S}$ called the structure tensor [8, 25, 30], These methods have produced excellent optical flow results [14].

Different noise models yield different estimators. TLS is a ML estimator when the noise in $\vec{o}_k$ is additive, isotropic and IID. When the noise is anisotropic and not identically distributed the formulation becomes much more complex [39]. More complex noise models, especially those with correlated noise in local regions, remain topics for future research.

9 Layered Motion: Mixture Models and EM

One common problem with area-based regression methods concerns the size of spatial support. With larger support there are more constraints for parameter estimation, but there is a greater risk that simple parametric motion models will be unsuitable. This is particularly serious near occlusion boundaries where multiple motions exist. For example, in the scene depicted in Fig. 6 the camera was translating, and therefore both the soda can and the background move with respect to the camera, but with different image velocities. To demonstrate this, Fig. 6 (right) shows a subset of the gradient constraints in the small region (marked in white) at the left side of the can. There are two points with a high density of constraint-line intersections, corresponding to the velocities of the can and the background.

One way to cope with regions with multiple motions is to explicitly model the *layers* in the scene. The layered model is like a cardboard cutout rep-

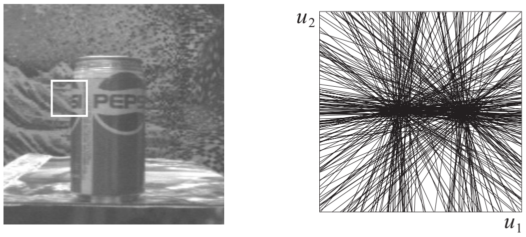

FIGURE 6. (left) The depth discontinuity at the left side of the can creates a motion discontinuity as the camera translates right. (right) Motion constraint lines in velocity space are shown from pixels within the white square. (After [31])

resentation of a scene in which different cardboard surfaces correspond to different layers, and they are assumed to be able to move independently [31, 49]. Layered motion estimation can be formulated using probabilistic mixture models, with the Expectation-Maximization (EM) algorithm for parameter estimation [4, 31, 53, 54].

Mixture Models

Let there be a region of pixels $\{\vec{x}_k\}_{k=1}^{K}$ in which we suspect there are multiple velocities. The region might contain an occlusion boundary for example. By way of notation, let $\vec{u}(\vec{x};\,\vec{c})$ denote a parameterized flow field with parameters $\vec{c}$. Within a single region of the image we will assume that there are N motions, parameterized by $\vec{c}_n$, for $1 \leq n \leq N$. Furthermore, according to the our *mixture model*, the individual motions occur with probability m_n. These *mixing probabilities* tell us what fraction of the K pixels within the region we expect to be consistent with (i.e., *owned* by) each motion. Of course the mixing probabilities sum to 1.

Let us further assume that we have one gradient constraint per pixel within the region. Let $\vec{o}_k \equiv (\nabla I(\vec{x}_k, t), I_t(\vec{x}_k, t))^T$ denote the spatial and temporal image derivatives at pixel $\vec{x}_k$. As above, given the correct motion, we assume that the gradient constraint is satisfied up to Gaussian noise:

$$e(\vec{x}_k;\,\vec{c}_n) \;\equiv\; \nabla I(\vec{x}_k, t)\cdot\vec{u}_n(\vec{x};\,\vec{c}_n) + I_t(\vec{x}_k, t) \;=\; \eta\ ,$$

where η is a mean-zero Gaussian random variable with a standard deviation of σ_v. Thus, the likelihood of observing a constraint $\vec{o}_k$ given the n^{th} flow model, is simply $p_n(\vec{o}_k \,|\, \vec{c}_n) = G(e(\vec{x}_k;\,\vec{c}_n);\,\sigma_v)$ where $G(e;\,\sigma)$ denotes a mean-zero Gaussian with standard deviation σ evaluated at e.

Finally, given the mixing probabilities and likelihood functions, the mixture model expresses the probability of a gradient measurement $\vec{o}_k$, as

$$p(\vec{o}_k \,|\, \vec{m},\, \vec{c}_1,\, ...,\, \vec{c}_N) \;=\; \sum_{n=1}^{N} m_n\, p_n(\vec{o}_k \,|\, \vec{c}_n)\ .$$

The probability of observing $\vec{o}_k$ is a weighted sum of the probabilities of observing $\vec{o}_k$ from each of the individual motions. The joint likelihood of a collection of K independent observations $\{\vec{o}_k\}_{k=1}^{K}$ is the product of the individual probabilities:

$$L(\vec{m}, \vec{c}_1, ..., \vec{c}_N) \;=\; \prod_{k=1}^{K} p(\vec{o}_k \,|\, \vec{m}, \vec{c}_1, ..., \vec{c}_N) \;. \tag{1.33}$$

Our goal is to find the mixture model parameters (the mixture proportions and the motion model parameters) that maximize the likelihood (1.33). Alternatively, it is often convenient to maximize the log likelihood:

$$\log L(\vec{m}, \vec{c}_1, ..., \vec{c}_N) \;=\; \sum_{k=1}^{K} \log\left(\sum_{n=1}^{N} m_n\, p_n(\vec{o}_k \,|\, \vec{c}_n)\right) \;.$$

EM and Ownerships

The EM algorithm is a general technique for maximum likelihood or MAP parameter estimation [12]. The approach is often explained in terms of a parametric model, some observed data, and some unobserved data. Our observed data are the gradient constraints, the model parameters are the motion parameters and mixing probabilities, and the unobserved data are the assignments of gradient measurements to motion models. Note that if we knew which measurements were associated with which motion, then we could solve for each motion independently from their respective constraints.

Roughly speaking, the EM algorithm is an iterative algorithm that iterates two steps that compute 1) the expected values of the unobserved data given the more recent estimate of the model parameters (the E Step), and then 2) the ML/MAP estimate for the model parameters given the observed data, and the expected values for the unobserved data.

A key quantity in this algorithm is called the *ownership probability*. An ownership probability, denoted $q_n(\vec{x}_k)$, is the probability that the n^{th} motion model is responsible for the constraint (i.e., generated the observed data) at pixel $\vec{x}_k$. This is an important quantity as it effectively segments the region, telling us which pixels belong to which motions. Using Bayes' rule, the probability that $\vec{o}_k$ is owned by model $\mathcal{M}_n$ can be expressed as

$$p(\mathcal{M}_n \,|\, \vec{o}_k) \;=\; \frac{p(\vec{o}_k \,|\, \mathcal{M}_n)\, p(\mathcal{M}_n)}{p(\vec{o}_k)} \;.$$

In terms of the mixture model notation here, this becomes

$$q_n(\vec{x}_k) \;=\; \frac{m_n\, p_n(\vec{o}_k \,|\, \vec{c}_n)}{\sum_{n=1}^{N} m_n\, p_n(\vec{o}_k \,|\, \vec{c}_n)} \;. \tag{1.34}$$

That is, the likelihood of the observation given the n^{th} model is simply $p_n(\vec{o}_k \mid \vec{c}_n)$, and the probability of the n^{th} model is just m_n. The denominator is the marginalization of the joint distribution $p(\vec{o}_k, \vec{c}_n)$ over the space of models. And of course it is easy to show that $\sum_n q_n(\vec{x}_k) = 1$. In the context of the EM algorithm these ownership probabilities can be viewed as soft assignments of data to models. Once these assignments are made we can perform a weighted regression to find the motion parameters of each model, using the same tools developed above for a single motion.

Given ownership probabilities, the updated mixing probability for model $\mathcal{M}_n$ is just the fraction of the total available ownership probability assigned to the n^{th} model, $m_n = \frac{1}{K}\sum_{k=1}^{K} q_n(\vec{x}_k)$. The estimation of the motion model parameters is similarly straightforward. That is, given the ownership probabilities, we estimate the motion parameters for each model independently as a weighted area-based regression problem. For the case of a translational motion model, where the motion parameters are just $\vec{c}_n \equiv \vec{u}_n$, this is just the minimization of the weighted least-squares error

$$E(\vec{u}_n) \;=\; \sum_{k=1}^{K} q_n(\vec{x}_k)\; [\, \nabla I(\vec{x}_k, t) \cdot \vec{u}_n + I_t(\vec{x}_k, t)\,]^2 \;\; . \tag{1.35}$$

Because the mixture model likelihood function (1.33) here will have multiple local minima, a starting point for the EM iterations is required. That is, to begin the iterative procedure one needs an initial guess of either the ownership probabilities, or of the model parameters (motion and mixture parameters). Often one starts by choosing random values for the initial ownership probabilities and then begin with the estimation of the mixing probabilities and the motion model parameters.

Outliers

As above, we must expect outliers among the gradient constraint observations. Gradient measurements near an occlusion boundary, for example, may not be consistent with either of the two motions. As a result, it is often extremely useful to introduce an outlier model, $\mathcal{M}_0$, in addition to the motion models; the likelihood for this outlier layer may be modeled with a uniform density [31]. Figure 7 shows results for the region near the can with two motion models and an outlier model like that described here. For the region shown in Fig. 7, the measurement constraints owned by the outlier model are shown in the bottom-right plot.

10 Conclusions

This chapter surveys several approaches to optical flow estimation. It is therefore natural to ask what works best? While historically some techniques have been shown to outperform others [6], in recent years several

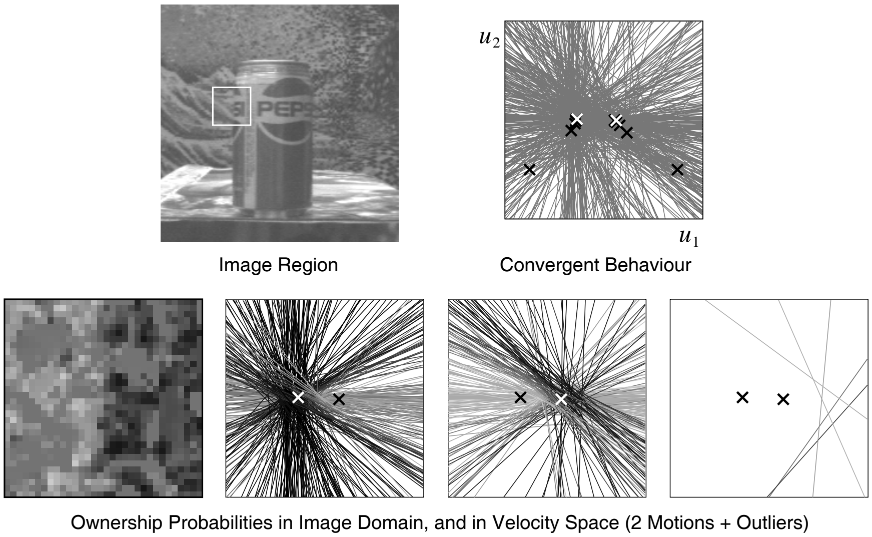

Image Region

Convergent Behaviour

Ownership Probabilities in Image Domain, and in Velocity Space (2 Motions + Outliers)

FIGURE 7. The top figures show a region at a depth discontinuity, and some of the constraint lines from pixels within that region. The black crosses in the upper right show a sequence of estimates at EM iterations. White crosses depict the final the estimates. The bottom figures showing ownership probabilities. The bottom left shows ownership probabilities at each pixel (based on the motion constraint at that pixel). The next two plots shown the velocity constraints where intensity depicts ownership (black denotes high ownership probability). The bottom right plot shows constraint lines owned by the outlier model. (After [31])

different approaches have produced excellent results on benchmark data sets, provided one pays attention to detail. Some of the important details include (1) multiple scales to help avoid local minima, (2) iterative warping and estimate refinement, and (3) robust cost functions to handle outliers. Accordingly, many techniques work well up to the limits of the key assumptions, namely, brightness constancy and smoothness.

Future research is needed to move beyond brightness constancy and smoothness. Detecting and tracking occlusion boundaries should greatly improve optical flow estimation. Similarly, prior knowledge concerning the expected form of brightness variations (e.g., given knowledge of scene geometry, lighting, or reflectance) can dramatically improve optical flow estimation. Brightness constancy is especially problematic over long image sequences where one must expect the appearance of image patches to change significantly. One promising area for future research is the joint estimation appearance and motion, with suitable dynamics for both quantities.

11 References

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2:284–299, 1985.

[2] E. H. Adelson and J. A. Movshon. Phonemenal coherence of moving plaid patterns. *Nature*, 300(5892):523–525, 1982.

[3] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[4] 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, 1995.

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

[6] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[7] J. R. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, pages 237–252. Springer-Verlag, 1992.

[8] J. Bigun, G. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, 1991.

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

[10] 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.

[11] A. Bruhn, J. Weickert, and C. Schnorr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society Series B*, 39:1–38, 1977.

[13] H. Farid and E. P. Simoncelli. Differentiation of discrete multidimensional signals. *IEEE Transactions on Image Processing*, 13(4):496–508, 2004.

[14] G. Farneback. Very high accuracy velocity estimation using orientation tensors, parametric motion models, and simultaneous segmentation of the motion field. In *IEEE International Conference on Computer Vision*, volume 1, pages 171–177, Vancouver, 2001.

[15] D. J. Fleet. *Measurement of Image Velocity*. Kluwer, Norwell, MA, 1992.

[16] D. J. Fleet, M. J. Black, Y. Yacoob, and A. D. Jepson. Design and use of linear models for image motion analysis. *International Journal of Computer Vision*, 36(3):169–191, 2000.

[17] 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.

[18] D. J. Fleet and A. D. Jepson. Stability of phase information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:1253–1268, 1993.

[19] 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.

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

[21] J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, Boston, 1950.

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

[23] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1025–1039, 1998.

[24] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Wiley, New York, 1986.

[25] H. Haussecker and D. J. Fleet. Estimating optical flow with physical models of brightness variation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):661–673, 2001.

[26] D. J. Heeger and A. D. Jepson. Subspace methods for recovering rigid motion i: Algorithms and implementation. *International Journal of Computer Vision*, 7(2):95–117, January 1992.

[27] B. K. P. Horn. *Robot Vision.* MIT Press, Cambridge, Massachusetts, 1986.

[28] B. K. P. Horn and B. G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[29] M. Irani and P. Anandan. Factorization with uncertainty. In *European Conference on Computer Vision*, pages 539–553, Dublin, 2000.

[30] B. Jähne, H. Haussecker, H. Spies, D. Schmundt, and U. Schurr. Study of dynamical processes with tensor-based spatiotemporal image processing techniques. In H. Burkhardt and B. Neumann, editors, *European Conference on Computer Vision*, pages 322–335, Freiburg, 1998. Springer.

[31] A. Jepson and M. J. Black. Mixture models for optical flow computation. In *Proc. IEEE Computer Vision and Pattern Recognition, CVPR-93*, pages 760–761, New York, June 1993.

[32] G. Li. Robust regression. In D. C. Hoaglin, F. Mosteller, and J. W. Tukey, editors, *Exploring Data Tables, Trends, and Shapes.* Wiley, 1985.

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

[34] B. D. Lucas and T. Kanade. An iterative image registration technique with an application in stereo vision. In *Seventh International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, 1981.

[35] R. Mann, A. D. Jepson, and J. M. Siskind. Computational perception of scene dynamics. *Computer Vision and Image Understanding*, 65(2):113–128, 1997.

[36] H. H. Nagel. On the estimation of optical flow: relations between different approaches and some new results. *Artificial Intelligence*, 33:299–324, 1987.

[37] S. Negahdaripour. Revised definition of optical flow: integration of radiometric and geometric clues for dynamic scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):961–979, 1998.

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

[39] O. Nestares and D. J. Fleet. Likelihood functions for general error-in-variables problems. In *IEEE International Conference on Image Processing*, page submitted, Barcelona, Spain, 2003.

[40] O. Nestares, D. J. Fleet, and D. J. Heeger. Likelihood functions and confidence bounds for total-least-squares estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 523–530, Hilton Head, 2000.

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

[42] C. Schnorr. Determining optical flow for irregular domains by minimising quadratic functionals of a certain class. *International Journal of Computer Vision*, 6(1):25–38, 1991.

[43] E. P. Simoncelli. *Distributed representation and analysis of visual motion.* PhD thesis, Department of Electrical Engineering, MIT, 1993.

[44] E. P. Simoncelli, E. H. Adelson, and D. J. Heeger. Probability distributions of optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 310–315, Mauii, 1991.

[45] M. Srinivasan, S. Zhang, M. Altwein, and J. Tautz. Honeybee navigation: Nature and calibration of the odometer. *Science*, 287(5454):851–853, 2000.

[46] R. Szeliski and J. Coughlin. Spline-based image registration. *International Journal of Computer Vision*, 22(3):199–218, 1997.

[47] S. Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society*, B-203:405–426, 1979.

[48] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cybernetics*, 60:79–97, 1989.

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

[50] W. H. Warren. Self-motion: Visual perception and visual control. In *Handbook of Perception and Cognition*, volume 5: Perception of Space and Motion. Academic Press, New York, 1995.

[51] A. B. Watson and A. J. Ahumada. Model of human visual-motion sensing. *Journal of the Optical Society of America A*, 2:322–342, 1985.

[52] J. Weber and J. Malik. Robust computation of optical-flow in a multiscale differential framework. *International Journal of Computer Vision*, 14(1):67–81, 1995.

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

[54] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 321–326, San Francisco, 1996.

[55] Y. Weiss and D. J. Fleet. Velocity likelihoods in biological and machine vision. In R. Rao, B. Olshausen, and M. Lewicki, editors, *Probabilistic models of the brain*. MIT Press, 2002.

[56] Y. Weiss, E. P. Simoncelli, and E. H. Adelson. Motion illusions as optimal percepts. *Nature Neuroscience*, 5(6):598–604, June 2002.