To appear in CVPR 97

Temporal Multi-scale Models for Flow and Acceleration

Yaser Yacoob and Larry S. Davis
Computer Vision Laboratory
Center for Automation Research
University of Maryland, College Park, MD 20742, USA

Abstract

A model for computing image flow in image sequences containing a very wide range of instantaneous flows is proposed. This model integrates the spatio-temporal image derivatives from multiple temporal scales to provide both reliable and accurate instantaneous flow estimates. The integration employs robust regression and automatic scale weighting in a generalized brightness constancy framework. In addition to instantaneous flow estimation the model supports recovery of dense estimates of image acceleration and can be readily combined with parameterized flow and acceleration models. A demonstration of performance on image sequences of typical human actions taken with a high frame-rate camera, is given.

1 Introduction

Image motion estimation involves relating spatial and temporal changes in image intensity to estimates of image flow. Articulated and deformable motions such as those encountered in images of humans in motion give rise to image sequences having, instantaneously, a wide range of flow magnitudes ranging from very small subpixel motions, whose recovery is inhibited by typical signal to noise constraints, to very large multiple pixel motions that can be recovered using expensive correlation methods or multi-resolution approaches. Here, we focus on the problem of estimating dense image flow for image sequences in which instantaneous flows range from 2-4 pixels/frame down to $1/16-1/32$ pixel/frame. The practical problem, of course, is that we do not know a priori which parts of the image are moving with which speed. Our solution is a scale-space like solution [11] in which we estimate image flow over a wide range of temporal scales, and combine these estimates (using both spatial and temporal constraints) using a combination of robust estimation and parametric modeling as in [5].

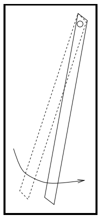


Figure 1: Pendulum movement illustrating varying velocities along its motion path

To motivate both the problem and our proposed solution consider a pendulum arm moving in front of a camera. The image flow will vary depending upon the distance of the measured point from the hanging point (see Figure 1). As we move towards the pendulum hanging point the instantaneous flow becomes very small and will fall in the noise range of the imaging system. As a result, two frame estimation and subsequent integration of these flow measurements over time will be highly noisy. In the context of human motion, the coincidence of lip motion with body and head motion, or the calf rotation around the knee create similar scale variations in the flow field.

The majority of published algorithms for estimation of image flow are based on two images (for a recent survey see [2]). Several approaches, however, consider the incremental estimation of flow [4, 13]; then, temporal continuity of the flow applied over a few images (for example, assuming constant acceleration) can improve the accuracy of the flow estimate. These approaches are based on computations between consecutive images. Other approaches use velocity-tuned filters (i.e., frequency-based methods) [8, 10] to compute the flow, and can be extended to flow estimation from several frames. The use of scale-space theory to compute optical flow was recently proposed by Lindeberg [12]. The proposed algorithm focused on scale selection in the spatial dimension so that different size im-

The support of the Defense Advanced Research Projects Agency (ARPA Order No. #C635), the Office of Naval Research (contract N000149510521) is gratefully acknowledged

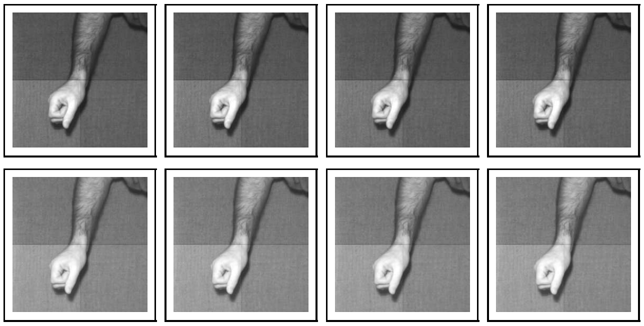


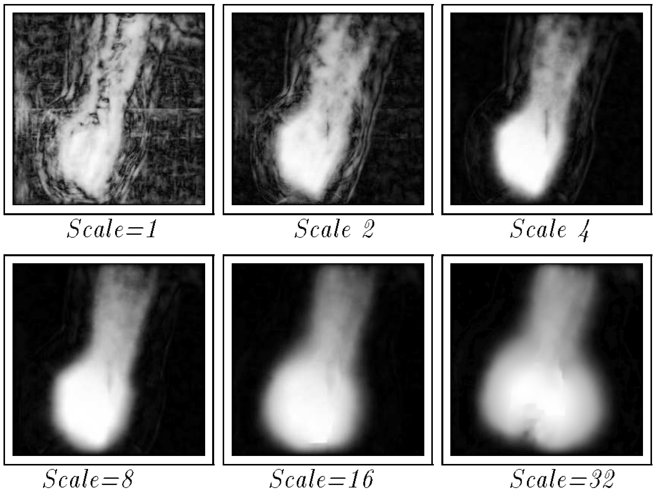
Figure 2: Eight images (each two frames apart) from a long sequence of a moving arm

age structures lead to different selection of scales for flow computation. The algorithm estimates flow from two images and involves spatial multi-scales.

2 A Motivating Example

We will use *scale*=1 to denote flow estimation between two consecutive images (i.e., the finest temporal resolution available), *scale* =2 to denote flow estimation between images that are two frames apart, etc. To illustrate the limitation of image flow estimates from any single scale we employ an image sequence of an arm moving in front of a camera. The sequence was taken with a high-frame-rate camera (500 frames per second) which allows us to capture the natural rapid motion of the arm. The arm (see Figure 2) is moving in a pendulum-like motion (with the hand rotating around the arm during the motion) in front of a lightly textured background*. Notice that there is a shadow created by the hand, leading to non-zero flow estimates of the shadow as well as the arm. The arm's intensity pattern consists of two parts: the arm itself is highly textured (allowing better flow estimation) while the hand is somewhat uniform in brightness. Figure 2 shows eight images from the sequence (chosen two frames apart). While the motion of the arm between two frames is very small, it will become apparent when the flow estimates are shown.

Figure 3 shows the image flow magnitudes for six scales (falling on a geometric scale 1,2,4,8,16, and 32 frames apart). The finest scale provides detailed estimates of the flow magnitude at the hand but quite noisy estimates along the arm, while the coarsest scale results in accurate estimates along the arm but considerably blurred and inaccurate estimates on the hand.


Scale=1 *Scale 2* *Scale 4*

Scale=8 *Scale=16* *Scale=32*

Figure 3: Enhanced flow display to show arm estimation at 1,2,4,8,16 and 32 scales.

3 A Multi-scale Flow Model

Let $I(x, y, t)$ be the image brightness at a point (x, y) at time t. The brightness constancy assumption at scale s is given by

$$I(x, y, t) = I(x + su\delta t, y + sv\delta t, t + s\delta t) \qquad (1)$$

where (u, v) is the horizontal and vertical image velocity at (x, y), δt is small. We assume, for now, that the instantaneous velocity (u, v) remains constant during the time span $s\delta t$ (leading to a displacement $(su\delta t, sv\delta t)$). This assumption is less likely to hold with the increase of scale and can lead to violations of brightness constancy. Let the range of scales over which flow is to be estimated be $1, .., n$. Expanding Equation (1) using a Taylor Series approximation (assuming locally constant flow) and dropping terms results in

$$0 = s({I^s}_x(x, y, t)u + {I^s}_y(x, y, t)v + {I^s}_t(x, y, t)) \qquad (2)$$

where I^s is the s-th frame (forward in time relative to I) of the sequence, and ${I^s}_x$, ${I^s}_y$ and ${I^s}_t$ are the spatial and temporal derivatives of image I^s relative to I.

Since Equation (2) is underconstrained for computation of (u, v), it is ordinarily posed as a minimization of a least squares error of the flow over a very small neighborhood, R, of (x, y), leading to

$$E(u, v, s) = \sum_{(x,y)\in R} (s({I^s}_x u + {I^s}_y v + {I^s}_t))^2 \qquad (3)$$

We have n equations of the form of Equation (3) one for each scale. The *scale-generalized* error is defined as

$$E_D(u, v) = \sum_{s=1}^{n} \sum_{(x,y)\in R} (s({I^s}_x u + {I^s}_y v + {I^s}_t))^2 \qquad (4)$$

*The quadrants' boundary intensity variation of the background is because the video-camera consists of four separate A/D banks. As a result, flow estimation at the quadrant boundaries is inaccurate. The problem could be overcome by local gain compensation.

Notice that Equation (4) biases the error term towards coarser scales due to the multiplication term s. Therefore, we normalize the error terms so that the minimization is in the form[†]

$$E_D(u,v) = \sum_{s=1}^{n} \sum_{(x,y)\in R} ({I^s}_x u + {I^s}_y v + {I^s}_t))^2 \qquad (5)$$

Equation (5) gives equal weight to the error values of all scales. Since it is expected that at each point (x,y) the accuracy of instantaneous motion estimation will be scale-dependent, we introduce a weight function $W(u,v,s)$ designed (see below) to minimize the influence of residuals of the relatively inaccurate scales. Equation (5) now becomes

$$E_D(u,v) = \sum_{s=1}^{n} \sum_{(x,y)\in R} (W(u,v,s)({I^s}_x u + {I^s}_y v + {I^s}_t))^2 \qquad (6)$$

Instead of the least squares minimization in Equation (6) we choose a robust estimation approach as proposed in [4], resulting in

$$E_D(u,v) = \sum_{s=1}^{n} \sum_{(x,y)\in R} \rho(W({I^s}_x u + {I^s}_y v + {I^s}_t), \sigma_e) \qquad (7)$$

where ρ is a robust error norm that is a function of a scale parameter σ_e. Since the weight function $W(u,v,s)$ should also reflect the degree of accuracy of the flow estimation we redefine it to include a scaling parameter σ_w, $W(u,v,s,\sigma_w)$. The choice of the weighting function W should satisfy the following constraints:

- It should take on values in the range $[0..c]$, c typically chosen as 1.0 for computational convenience.
- For a large σ_w, W should approach 1.0 regardless of (u,v) and s.
- Given σ_w, larger estimated flow (u,v) at point (x,y) should lead to higher weights for the lower scales of the error term ${I^s}_x u + {I^s}_y v + {I^s}_t$, while a small flow should lead to higher weights of the highest scales.

Figure 4 reflects qualitatively the desired shape of the weighting function for a fixed σ_w. It illustrates the weighting as a function of scale s and flow magnitude $||(u,v)||$ at (x,y). The following Gaussian function satisfies the above requirements

$$W(u,v,s,\sigma_w) = e^{-(s - \frac{n}{(\alpha||(u,v)||^2 + 1.0)})^2 / 2{\sigma_w}^2} \qquad (8)$$

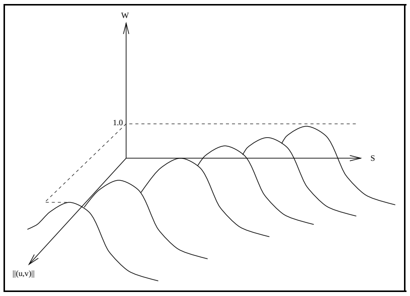


Figure 4: The weighting function as a function of s and flow magnitude $||(u,v)||$

where $||(u,v)||^2$ is the squared magnitude of the current flow estimate at (x,y), and α is a constant. Notice that when $||(u,v)||^2 << 1.0$ the maximal weight occurs at the highest scale n, while higher values of $||(u,v)||^2$ lead to a maximal weight at lower scales; specifically the Gaussian is centered at $\frac{n}{\alpha||(u,v)||^2+1.0}$. The scale parameter σ_w determines the width of the Gaussian, and the constants α and 1.0 can be changed to further shift the maximal weight scale location. The application of the weighting function in the estimation is as follows: in the first iteration, all scales are given equal weight (1.0) by selecting a large σ_w. Afterwords, iteratively, the estimates are refined by decreasing σ_w.

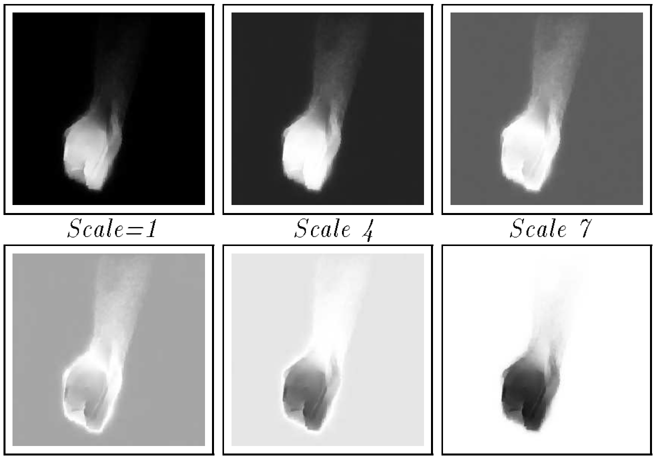
This temporal multi-scale procedure is accompanied by a spatial coarse-to-fine strategy [3] that constructs a pyramid of the spatially filtered and sub-sampled images (for more information see [4]) and computes the flow initially at the coarsest level and then propagates the results to finer levels. The computational aspects of the multi-scale model follow, generally, the approach proposed by Black and Anandan [4, 5].

4 Experimental Results

In the following figures we show the results of image flow computation when $\sigma_w = 20.0$ and is decreased at a rate of 0.85 for five iterations, and $\sigma_e = 100.0$ and is decreased also at a rate of 0.85 for 40 iterations. The computation is performed over 16 scales.

Figure 5 illustrates the weights at several scales during the computation of image flow (the brighter the intensity the higher the weight; weights across scales were normalized in these images to allow for comparisons). At *scale*=1 only the hand area is given high weights while the arm and the background are given very low weights. As the scale increases the weights are increased along the arm and the background while a decrease on the hand gradually takes place. At the highest scale (*scale* = 16) the hand's weight is very low while the arm and the background receive a high weight. Figure 6 shows the effect of the iterative re-

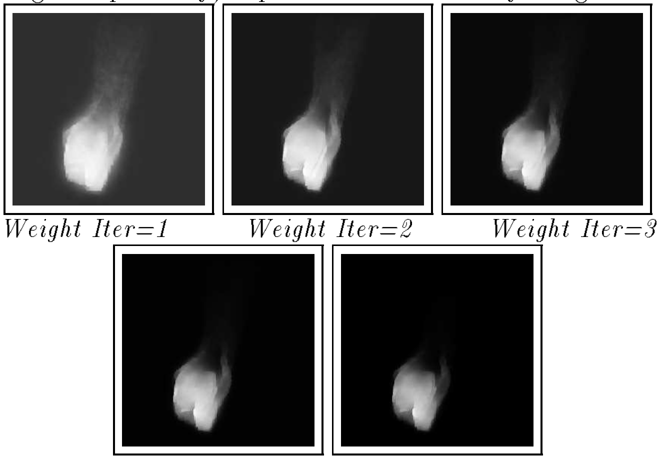
[†] The same effect could have been achieved by dividing the right side of Equation (2) by s for all scales prior to error summation.



Scale=1 *Scale 4* *Scale 7*

Scale=10 *Scale=13* *Scale=16*

Figure 5: The weighting function W as computed at the scales 1,4,7,10,13 and 16 scales (top left to bottom right respectively) expressed as an intensity image.



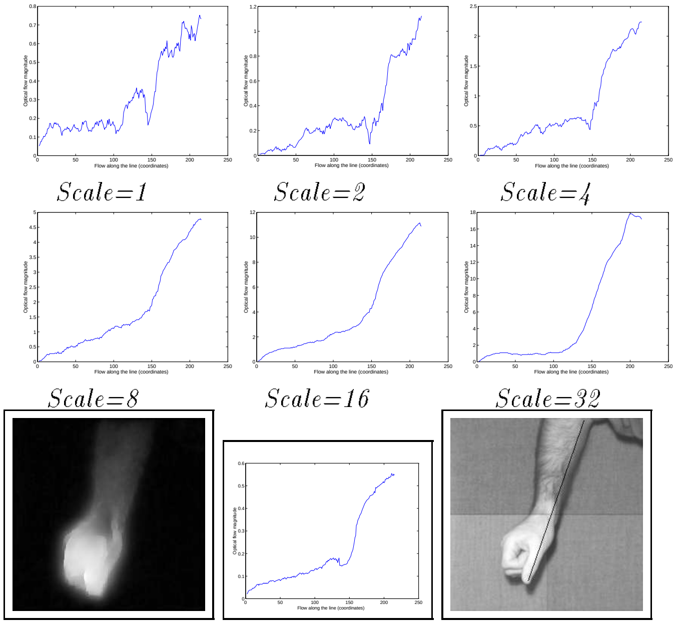
Weight Iter=1 *Weight Iter=2* *Weight Iter=3*

Weight Iter=4 *Weight Iter=5*

Figure 6: The weighting function W at scale 1 (finest scale) as evolved in five iterations

finement of the weighting function W for *scale* = 1 (the finest scale) on the relative weights for different regions. The values are normalized across the five images to allow comparison. Notice that the first iteration gives high weights to the hand, and the weights given to the arm and the background are somewhat significant. The fifth iteration also gives high weights to the hand while the arm and the background have the lowest weight, and they are much lower than after the first iteration.

Figure 7 (top and middle rows) shows graphs of the individual scale flow magnitudes computed along a line drawn down the center of the arm (bottom right). These graphs correspond to the scale computations shown in Figure 3. Since the arm is *approximately* moving like a pendulum with the hand simultaneously rotating around the wrist (see Figure 10), the flow should



Scale=1 *Scale=2* *Scale=4*

Scale=8 *Scale=16* *Scale=32*

Figure 7: The flow magnitude along a line (bottom right) computed using a single scale ($s = 1, 2, 4, 8, 16$ and 32 scales; top and middle rows), the multi-scale flow magnitudes (bottom left), and the multi-scale flow magnitudes along the line (bottom center)

increase slowly along the arm then jump considerably on the hand. This is clearly visible in these graphs. The dip in these graphs (occurring between 125-145) is a result of the intensity discontinuity of the four quadrants of the camera. Figure 7 also shows the multi-scale flow magnitude results (bottom left). The flow boundary is quite sharp and the corresponding flow magnitude along the line is shown (bottom center); it measures a very smooth change in the flow along the arm and significant increase at the hand (with maximal flow at the finger).

In order to compare the performance of single scale ($scale = 1$) and multi-scale flow estimation, we generated a sequence of images using a synthetic flow model where we have ground-truth data. Figure 8 (top) shows an image of a person during a walking activity. The synthetic sequence is generated by warping the image patch of the "calf" foreward according to a multi-scale parameterized motion model for several frames (assuming constant velocity). The estimated multi-scale (12 scales) flow magnitudes are shown (top right). A quantitative comparison along a line on the "calf" between the original flow (bottom, solid line) the single scale flow (dotted line) and the multi-scale (dashed line). The multi-scale estimate is closer to the synthetic flow than the single scale estimation. Accurate recovery of the flow is actually limited by interpolation side effects

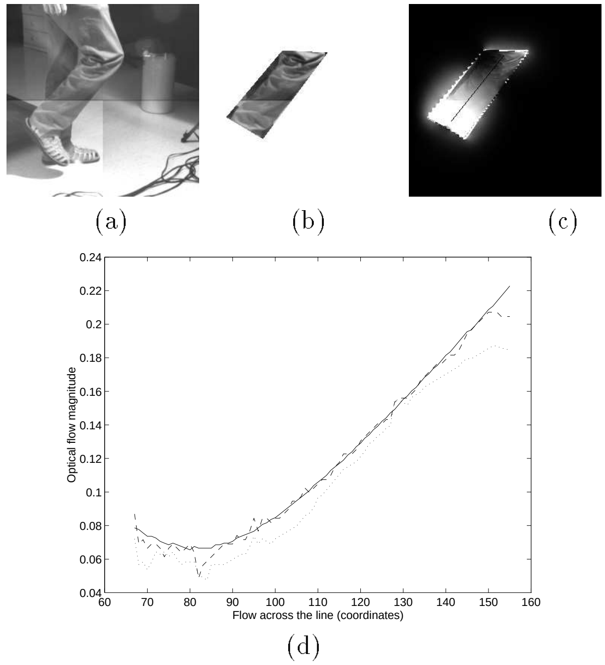


Figure 8: A synthetic motion example that compares flow magnitudes on a real image of a calf. The image (see (b)) was warped and the flow magnitudes along a line (see (c)) are shown as a solid line (see (d)). The estimates of flow magnitudes using 1 and 12 scales over the same line are shown ((d), dotted and dashed lines, respectively).

in generating the synthetic motion.

5 Estimation of Image Acceleration

The scale-generalized brightness constancy assumption given in Equation (1) assumes constant flow at all scales. This can be extended to include acceleration models. Let the image flow as a function of scale s be $(u(s), v(s))$. Then the brightness constancy assumption at scale s becomes

$$I(x,y,t) = I(x+\sum_{s} u(s)ds, y+\sum_{s} v(s)ds, t+s) \quad (9)$$

As a special case, if image motion is assumed to be subject to a constant acceleration, the flow can be given by

$$u(s) = x_0 + x_1 s \quad (10)$$

$$v(s) = x_2 + x_3 s \quad (11)$$

where x_1 and x_3 are the horizontal and vertical acceleration terms. Note that in the context of a long sequence this model supports a piecewise constant acceleration assumption. If acceleration fluctuations within the scales involved in the estimation are small or fall within the performance range of the robust estimator

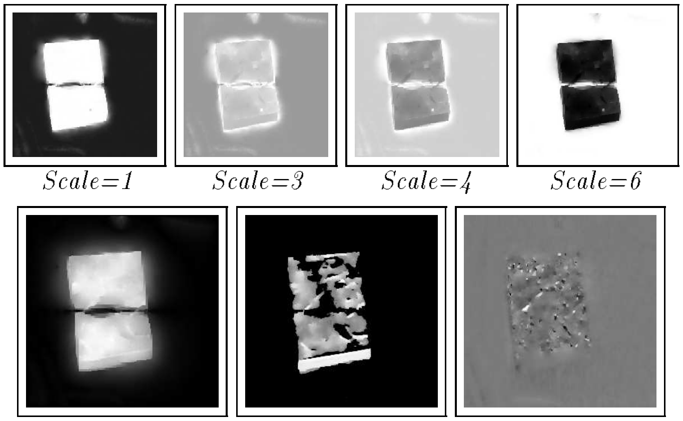


Figure 9: The weights (upper row) at scales 1, 3, 4 and 6, respectively (out of 6 scales), and the flow magnitude and vertical and horizontal accelerations (bottom row, left to right, respectively) for a falling book.

(about 35%-40% outliers) this model holds. This flow model leads to a brightness constancy assumption of the form

$$I(x,y,t) = I(x+\sum_{i=1}^{s}(x_0+x_1 i), y+\sum_{i=1}^{s}(x_2+x_3 i), t+s) \quad (12)$$

Using a Taylor Series expansion and dropping terms (including scale normalization) we arrive at

$$0 = {I^s}_x(x_0+x_1\frac{s+1}{2}) + {I^s}_y(x_2+x_3\frac{s+1}{2}) + {I^s}_t \quad (13)$$

The new scale-generalized error function is given by

$$E_D(u,v) = \sum_{s=1}^{n}\sum_{(x,y)\in R}\rho(W({I^s}_x(x_0+x_1\frac{s+1}{2}) + \quad (14)$$

$$\qquad {I^s}_y(x_2+x_3\frac{s+1}{2})) + {I^s}_t), \sigma_e)$$

Figure 9 shows the dense flow and acceleration estimated for a book-falling sequence (see also Figure 11). The top row shows the the weighting function's values assigned for each scale (normalized to enhance the contrast). At low scales the book's region is assigned high weight while the background is assigned very low weight. This is reversed as scale is increased, so at the top scale the motion of the book is so large that little weight is given to the book area. The bottom row shows the dense velocity magnitude (left) and the vertical and horizontal accelerations (center and right, respectively). Notice that the estimated horizontal acceleration is almost uniformly zero.

6 Parameterized Flow Models

Dense flow computation generates large data sets that may not be easily used in higher level vision tasks. Recently, it has been demonstrated that parameterized flow models can provide a powerful tool for reasoning about image motion between successive images (see [6]). The multi-scale flow estimation algorithm can be extended in a straightforward way to parameterized models of image flow. In this section we describe the extension of the muti-scale framework to affine and planar parameterized image motion models.

Recall that the flow constraint given in Equation (2) assumes constant flow over a small neighborhood around the point (x, y). Over larger neighborhoods, a more accurate model of the image flow is given by low-order polynomials [1]. For example, affine motion is given by

$$U(x, y) = a_0 + a_1 x + a_2 y \tag{15}$$

$$V(x, y) = a_3 + a_4 x + a_5 y \tag{16}$$

where a_i's are constants and (U, V) is the instantaneous velocity vector. Equation (7) now becomes

$$E_D(U, V) = \sum_{s=1}^{n} \sum_{(x,y)\in A/P} \rho(W(U, V, s, \sigma_w)(I^s{}_x U + I^s{}_y V + I^s{}_t), \sigma_e) \tag{17}$$

where A/P denotes the region in which the flow is assumed to be affine (A) or planar (P). The minimization of Equation (17) results in estimates for the parameters a_i. The choice of the weighting function W is somewhat more complex here than it was previously. The weighting function can be designed using the current flow estimates computed by the model (U, V). This weighting leads to different weights within the region according to the magnitude of the flow so that at points where the flow estimate is low the coarser scales will be more dominant while the larger flow estimates will determine the finer scales. Alternatively, W can be designed using the parameters of the model a_i (i.e., $W(\bar{a}, s, \sigma_w)$ where $\bar{a}$ is the set of model parameters). The former leads to a computation based on weighting of spatio-temporal derivatives while the latter leads to weighting of parametric models. Once a choice for the weighting function has been made the computation of the parameters of the model follows the approach proposed in [4].

In the examples in this chapter we adopt the weight of parametric models. Recall that the parameters of the affine and planar models capture several aspects of the region's motion (see [6]). Since the translation of the region is of most interest the parameters a_0 and a_3 can be substituted as $||(a_0, a_3)||$ for $||(u, v)||$ in Equation (8).

Figure 10 shows the results of parameterized flow estimation over the hand region of the moving arm over a long sequence (about 540 frames). The parameterized flow is used to automatically track the hand

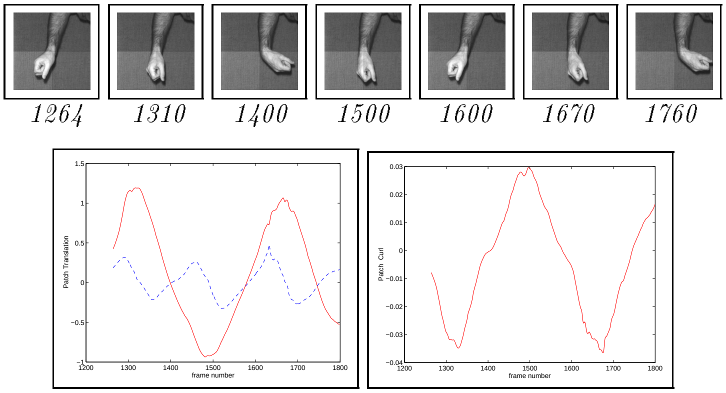


Figure 10: Seven images of a long sequence of the arm in motion (top).Flow results without acceleration model, a_3 and a_0 (left, solid and dashed lines, respectively) and curl (center row left to right).

motion throughout the sequence similar to [6] (assuming an initial manual hand segmentation in the first image). The frame numbers are shown with the images. The left graph shows the horizontal and vertical translations (solid and dashed lines, respectively) and the right graph shows the *curl* of the hand. Notice the smoothness and robustness of these figures.

Parameterized flow models can also be extended to include acceleration. The extension of the affine model requires that the motion parameters across scales be dependent on the scale so that a_i becomes $a_i(s)$. Assuming a constant acceleration for these parameters, the models now become

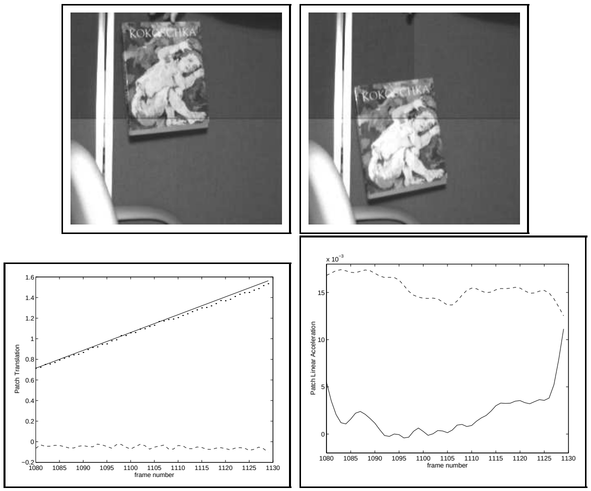
$$U(x, y) = (a_0 + a_0{}'s) + (a_1 + a_1' s)x + (a_2 + a_2' s)y \tag{18}$$

$$V(x, y) = (a_3 + a_3{}'s) + (a_4 + a_4' s)x + (a_5 + a_5' s)y \tag{19}$$

where a_0', a_3' are the linear horizontal and vertical acceleration components of the motion and the a_1', a_2', a_4' and a_5' are acceleration components that can be related to angular, divergence and deformation accelerations.

Figure 11 describes an experiment in which the acceleration of a falling book is estimated from an image sequence.[‡] Notice that although the book is falling vertically, a small horizontal motion component is present (observe the change of the upper left corner of the book relative to the white stripes). The bottom left graph of Figure 11 shows the horizontal and vertical velocity computed for the sequence (dashed and dotted lines, respectively), and the *predicted* vertical velocity (solid line) based on the velocity computed at the first frame and the *average* acceleration in the first ten frames. The graphs suggest that the inclusion of acceleration

[‡] The book is manually segmented in the first image and tracked automatically afterwords using our multi-scale parameterized flow model.



(a) (b)

Figure 11: Two images (60 frames apart) of a falling book, the vertical and horizontal velocities (dashed and dotted lines, respectively (a)), and the predicted vertical velocity from the starting velocity and average acceleration, a_3', in the first 10 frames (solid line, (a)) and the vertical and horizontal accelerations (dashed and solid lines respectively, (b))

in the image motion model is valuable in predicting the real motion for a significant amount of time.

7 Discussion

The proposed multi-scale approach for computing optical flow and acceleration introduced explicit temporal models for image intensity and flow changes. As demonstrated in several image sequences, a multi-scale framework can increase the accuracy of instantaneous motion estimates and recover simultaneously both flow and acceleration.

Algorithms for motion estimation can be quite noisy since they are based on local operators applied over very small neighborhoods between two images. Temporal smoothing was proposed by [5] in a regularization framework; in contrast our multi-scale approach employs well-understood scale-space concepts [11, 12] to create smooth estimates. Due to the integrative nature of the multi-scale estimation, motion smoothing is achieved through the estimation process.

In this paper we developed a new multi-temporal framework for computing flow and acceleration in images. Both dense and parameterized representations were employed and demonstrations on long image sequences were provided. This approach is an extension of the popular brightness-constancy assumption to a temporal scale-space domain. It provides for higher accuracy over a wider range of flows in image sequences.

Acknowledgements

We would like to thank Michael Black for providing the code for optical flow estimation from two images.

References

[1] Adiv G. *Determining three-dimensional motion and structure from optical flow generated by several moving objects.* IEEE PAMI, Vol. 7(4), 1985, 384-401.

[2] S.S. Beauchemin and J.L. Barron. *The Computation of Optical Flow.* ACM Computing Surveys, Vol. 27, No. 3, 1995, 433-467.

[3] J.R. Bergen, P. Anandan, K.J. Hanna and R. Hingorani. *Heirarchical model-based motion estimation.* In G. Sandini, editor, ECCV-92, Vol. 588 of LNCS-Series, 237-252, Springer-Verlag, 1992.

[4] M.J. Black and P. Anandan. *A Frame-work for Robust Estimation of Optical Flow.* ICCV 1993, 231-236.

[5] M.J. Black and P. Anandan. *The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields.* 1994 Revision of Technical Report P93-00104, Xerox PARC, December 1993.

[6] M.J. Black and Y. Yacoob. *Recognizing facial expressions in image sequences using local parameterized models of image motion.* ICCV, 1995, 374-381.

[7] A. Blake and A. Zisserman. *Visual Reconstruction* The MIT Press, Cambridge, Massachusetts, 1987.

[8] D.J. Fleet and A.D. Jepson. *Computation of Component Image Velocity from Local Phase Information.* IJCV, Vol. 3, No. 4, 77-104.

[9] S. Geman and D.E. McClure. *Statistical Methods for Tomographic Image Reconstruction.* Bulletin of the International Statistical Institute, LII-4:5-21, 1987.

[10] D.J. Heeger. *Optical Flow Using Spatio-temporal Filters.* IJCV, Vol. 1, 279-302.

[11] T. Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer Academic Publishers, 1994.

[12] T. Lindeberg. *A Scale Selection Principle for Estimating Image Deformations.* Technical Report, Stockholm University, CVAP 196, 1996.

[13] A. Singh. *Incremental Estimation of Image Flow Using a Kalman Filter.* IEEE Proceedings of the Workshop on Visual Motion 1991, 36-43.