Key terms: Motion Estimation, Optical Flow, Non-rigid Motion.

Yaser Yacoob

Computer Vision Laboratory

University of Maryland, College Park, MD 20742, USA

yaser@umiacs.umd.edu

Footnotes

Yaser Yacoob

Computer Vision Laboratory

University of Maryland, College Park, MD 20742, USA

- The quadrants' boundary intensity discontinuity a result of of the video-camera consisting 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.
- The dense flow algorithms of Black and Anandan [6] is used. The algorithm's parameters were changed to achieve best measurement results.
- The same effect could have been achieved by dividing the right side of Equation (3) by s for all scales prior to error summation.
- The book is manually segmented in the first image and tracked automatically afterwords using our multi-scale parameterized flow model.

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 temporal changes in image intensity across the spatial dimensions. 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 sub-pixel 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. Table 1 shows the categories of image motion, typical motion values that can be recovered

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

Category	Motion amount m	Model/Algorithm
Very large	$> R$	Correlation
Large	$1.0 < m < R$	Pyramid+ Image Gradients, Correlation or Motion Filters
Small	$\sigma << m < 1.0$	Image Gradients or Correlation
Very small	$\sigma \approx m$	?? *Multi-frame*

Table 1: Categories of image motion, amount of motion m (pixels/frame) and potential algorithms. R is the search radius around the point (e.g., in a correlation-based algorithms) and is equal to 2^p (p is the number of levels in the case of pyramid-based methods) for differential methods. The noise level is σ (a standard deviation in the case of Gaussian noise).

and the computational algorithms currently available for their estimation. Differential methods perform well at estimating small motions, and in conjunction with a coarse-to-fine strategy they can estimate large motions; but they fail when motions are very large. Correlation approaches are better suited to compute very large motions but they are computationally expensive. Estimating very small motions is challenging since sufficient (but unknown) time has to elapse before the motion can be reliably estimated. Too small a temporal window makes motion estimation highly susceptible to noise while too large a temporal window might drive the estimation outside the domain of a differential algorithm.

Here, we focus on the problem of estimating dense image flow for image sequences in which instantaneous flows range from very small to large and these flows occur simultaneously at different locations in the image. 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 [13] in which we estimate image flow over a range of temporal scales, and combine these estimates (employing both spatial and temporal constraints) using a combination of robust estimation and parametric modeling.

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 point from the hanging point (see Figure 1, left). As we move towards the pendulum hanging point the instantaneous flow becomes very small and its measurement at small temporal scale suffers from low signal to noise ratio. As a result, two frame estimation and subsequent integration of these flow measurements over time will be inaccurate.

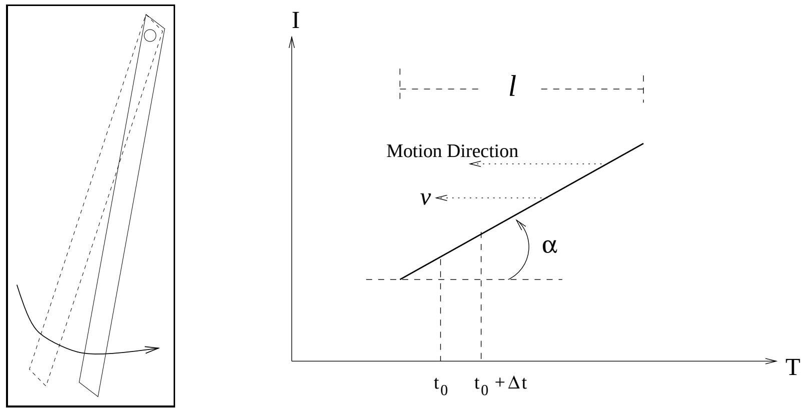


Figure 1: Left, pendulum movement illustrating varying velocities along its motion path. Right, a ramp with inclination $\tan(\alpha)$ in 1D motion

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.

To appreciate, quantitatively, the difficulty of computing small motions, consider the simpler problem of estimating the motion of a 1D ramp (which is equivalent to estimating the ramp's inclination $\tan(\alpha)$) as a function of time lapse Δt and velocity v (see Figure 1, right). We make the simplifying assumption that all noise sources can be modeled by a normal distribution with zero mean and standard deviation, σ_1, (for a discussion of noise sources in flow estimation see [12, 16]). Under this assumption it can be easily shown that the estimated inclination, $\tan(\beta)$, satisfies the following if the measurement errors are less than or equal to the standard deviation $\pm\sigma$:

$$\tan(\alpha) - \frac{2\sigma_1}{v\Delta t} \leq \tan(\beta) \leq \tan(\alpha) + \frac{2\sigma_1}{v\Delta t} \tag{1}$$

Equation (1) indicates that as long as $\tan(\alpha) >> \frac{2\sigma_1}{v\Delta t}$ the estimation error (i.e., $\tan(\beta) - \tan(\alpha)$) remains very small. However, this error increases if either $\tan(\alpha)$ becomes very small or $v\Delta t$ becomes very small (e.g., due to a small v while Δt is kept constant). The former case is, in fact, the aperture problem in 1D while the latter case has not been, so far, addressed in image motion research. This error is inversely proportional to the time lapse Δt (assuming σ_1 and v are held constant). Clearly, to reduce the estimation error it is beneficial to choose a $\Delta t >> \frac{2\sigma_1}{v}$, which favors long time lapses. However, since the linear ramp has only a finite extent, l, the maximal Δt has to be less than l divided by v (see discussions on sine wave motion in [2, 11, 12, 16]).

It is worth noticing that since current image motion estimation techniques use two frames ($\Delta t = 1$), the estimation error is high unless $\tan(\alpha) >> \frac{2\sigma_1}{v}$. Hence, no matter what the value of σ_1 is, there is a velocity, v, for which the estimation error is unacceptably high. Also, regardless of the algorithm used to estimate $tan(\beta)$, Equation (1) illustrates the theoretical bounds on accuracy. This 1D example can be generalized to 2D surfaces in motion where the fundamental trade-off between estimation accuracy, on the one hand and time lapse and spatial frequency, on the other hand, also holds.

The majority of published algorithms for estimation of image flow are based on two images (for a recent survey see [3]). Several approaches, however, consider the incremental estimation of flow [5, 15]; 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, however, are still based on computations between consecutive images. Other approaches use velocity-tuned filters (i.e., frequency-based methods) [9, 11] to compute the flow, and can be extended to flow estimation from several frames, perhaps employing a multi-scale approach similar to what is described here. The use of scale-space theory to compute optical flow was recently proposed by Lindeberg [14]. The proposed algorithm focused on scale selection in the spatial dimensions so that different size image structures lead to different selection of scales for flow computation. The algorithm, however, estimates flow from two images and involves only the spatial multi-scales.

We develop a method for computing optical flow taking into consideration the fact that motions may be most accurately estimated at different temporal scales. We assume that no a priori information on the spatio-temporal distribution of flow values is available, thus allowing our method to be used with any

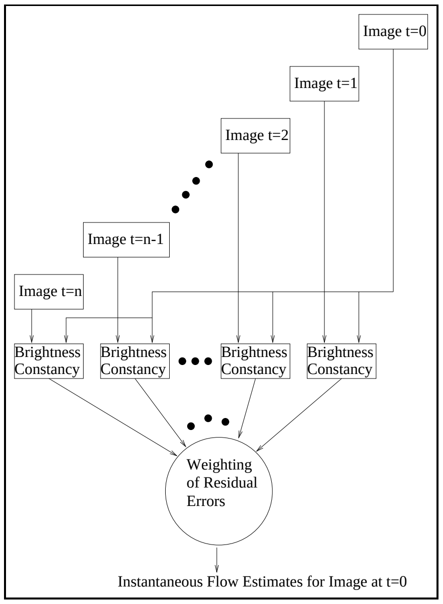


Figure 2: A diagram of the multi-scale approach

image sequence. This method can also be viewed as a multi-frame flow estimation when the utility of all scales is not equal. We discuss how to extend the approach to estimation of image acceleration and to the measurement of parameterized flow models; we provide demonstrations on several image sequences.

Figure 2 illustrates the approach we propose for computing image flow from multiple frames. Given an image at time $t = 0$ our objective is to compute image motion estimates based on subsequent images $t = 1, ..., $n. We assume that brightness constancy holds and measure its satisfaction as a constraint on flow estimation. The errors of all brightness constancy constraints are weighed according to a measure of accuracy and the minimum of sum of errors is considered the best estimate.

The paper is organized as follows. Section 2 illustrates, using an image sequence, the inadequacy of single scale flow estimation. In Section 3 we describe the motion model employed for estimating image flow from multiple scales. This is followed by experimental results in Section 4. Section 5 provides the extension of the model to compute image acceleration. In Section 6 we describe the estimation process when parameterized motion models are used. Finally, in section 7 a discussion and summary of our approach are provided.

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 3) is rotating in a pendulum-like motion with an additional rotation of the hand around the wrist during its 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. Figure 3 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. The correct instantaneous flow should show a spatially coherent flow in which the flow increases as we move from the upper part of the arm toward the hand and then significantly increases on the hand.

Figure 4 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 better estimates along the arm but considerably blurred estimates on the hand.

Figure 5 is a rescaled version of Figure 4 in which the small flow values along the arm can be more easily observed. The flow estimation along the arm at fine scales is dominated by the noise of the imaging system. As scale increases, better estimates are computed along the arm at the cost of blurring the flow of the hand. As a consequence, if motion segmentation into parts is sought, the finest scale would result

*The quadrants' boundary discontinuity a result of of the video-camera consisting 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.

†The dense flow algorithms of Black and Anandan [6] is used. The algorithm's parameters were changed to achieve best measurement results.

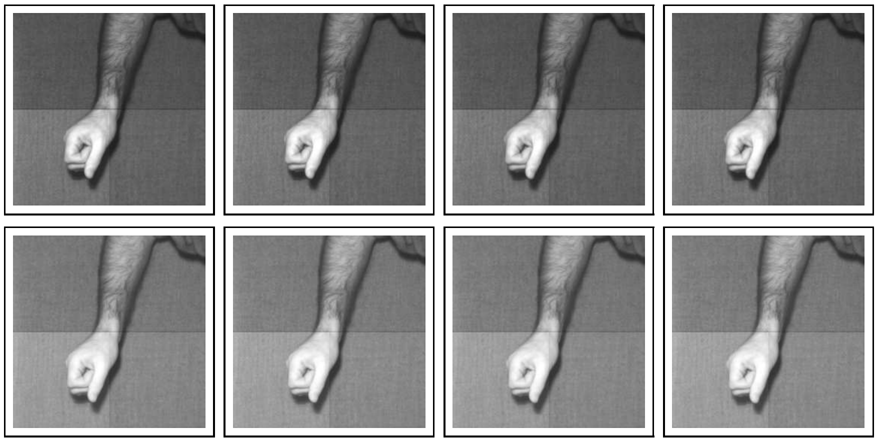


Figure 3: Eight images (each two frames apart) from a long sequence of a moving arm

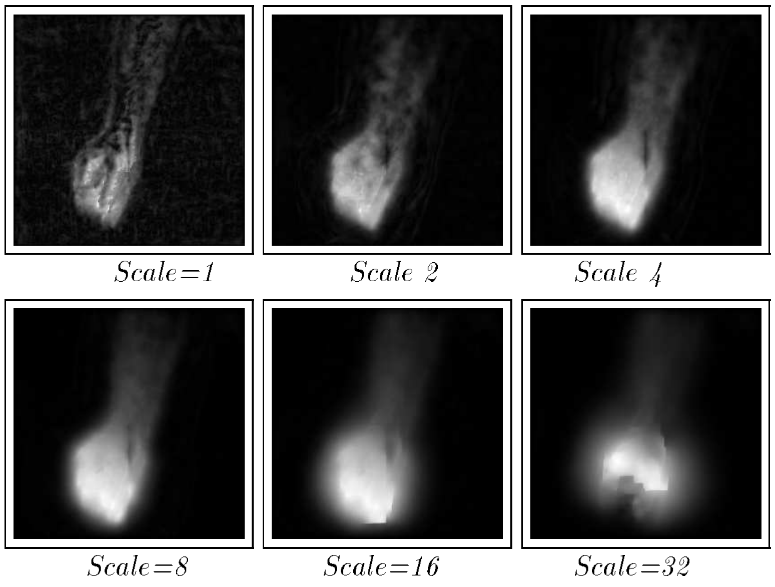


Figure 4: Flow magnitudes at 1,2,4,8,16 and 32 scales (top left to bottom right respectively)

in highly fragmented components, while the coarsest scale would lead to highly inaccurate boundaries for the hand.

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

$$I(x, y, t) = I(x + su\delta t, y + sv\delta t, t + s\delta t) \tag{2}$$

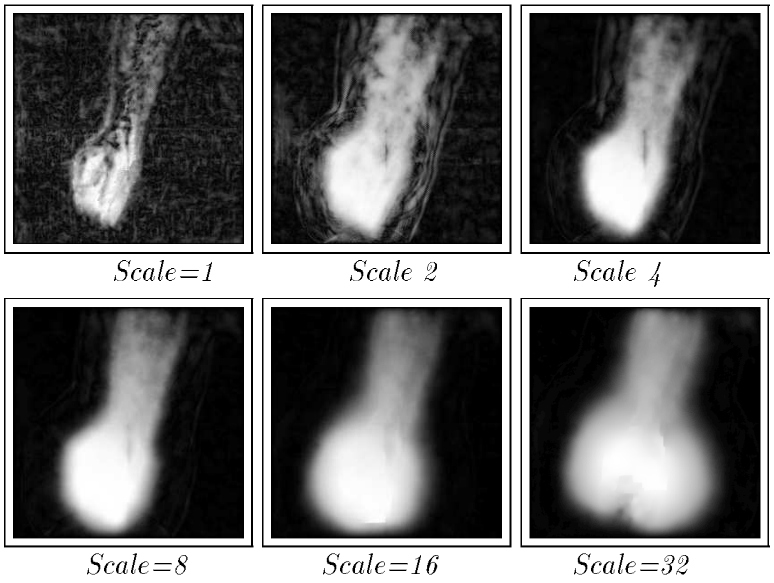


Figure 5: Enhanced flow display to show arm estimation at 1,2,4,8,16 and 32 scales (top left to bottom right respectively)

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 (2) using a Taylor Series approximation (assuming locally constant flow) and dropping terms results in

$$0 = s({I^s}_x(x, y, t)u + {I^s}_y(x, y, t)v + {I^s}_t(x, y, t)) \tag{3}$$

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 (3) is underconstrained for recovery of (u, v), it is ordinarily posed as a minimization of a robust error norm of the flow over a very small neighborhood, R, of (x, y), leading to

$$E(u, v, s) = \sum_{(x,y)\in R} \rho(s({I^s}_x u + {I^s}_y v + {I^s}_t), \sigma_e) \tag{4}$$

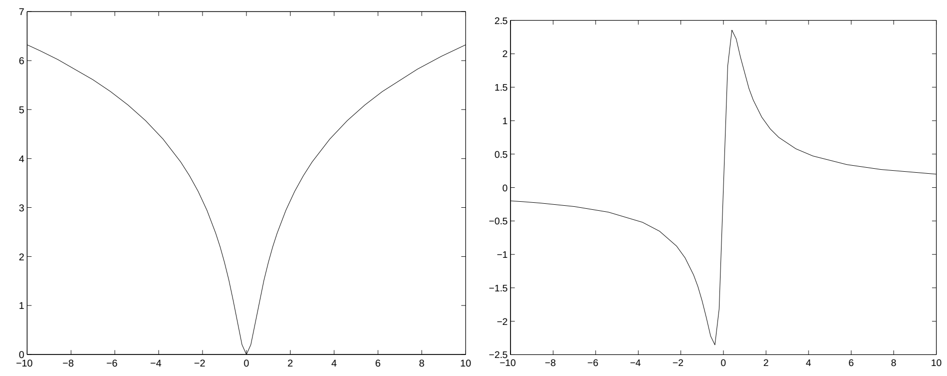


Figure 6: An illustration of the Lorentzian function ρ and its derivative ρ'

where ρ is a robust error norm [10] that is a function of a scale parameter σ_e. In our implementation we use the Lorentizian function and its derivative (see Figure 6),

$$\rho(x, \sigma_e) = \log(1 + \frac{1}{2}(\frac{x}{\sigma_e})^2) \quad \text{and} \quad \rho'(x, \sigma_e) = \frac{2x}{2{\sigma_e}^2 + x^2}. \tag{5}$$

The parameter σ_e determines the point at which the influence of a measurement begins to diminish. All points with absolute error greater that $2\sqrt{2}\sigma_e$ are considered outliers (more details can be found in [6, 10]).

We have n equations of the form of Equation (4) one for each scale. The *scale-generalized* error is defined as

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

Notice that Equation (6) 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[‡]

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

[‡]The same effect could have been achieved by dividing the right side of Equation (3) by s for all scales prior to error summation.

Equation (7) 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 error terms of the relatively inaccurate scales. Equation (7) now becomes

$$E_D(u, v) = \sum_{s=1}^{n} \sum_{(x,y)\in R} \rho(W(u, v, s)(I^s{}_x u + I^s{}_y v + I^s{}_t), \sigma_e). \tag{8}$$

Before discussing the design choices for the weighting function W, it is worth considering quantitatively the residual error values at each scale. Consider again the 1D ramp example discussed in Section 1, the inclination error assuming noise within $\pm\sigma$ is given by

$$E_T = \tan(\beta) - \tan(\alpha) \leq \pm\frac{2\sigma_1}{v\Delta t}. \tag{9}$$

Therefore, given a Gaussian noise model, the estimation error will be Gaussian with standard deviation $\pm\frac{2\sigma_1}{v\Delta t}$. At the same time, the physical extent of the ramp determines the maximal time over which accurate derivative estimates are computable. The best-case scenario occurs when the initial measurement occurs at the beginning of the ramp. In this case, the maximal time delay is given as $\Delta t < l/v$ where l is the projected length of the ramp (i.e., $l = r\cos(\alpha)$ where r is the full length of the ramp). As the initial measurement approaches the end of the ramp there is diminishing time to obtain a second measurement on the same ramp.

Figure 7 shows the error functions of Equation (9) for different v's. The red, green and blue graphs show the error, E_T, for three speeds $v = 0.5, 0.05, 0.025$ pixels/frame and $\sigma_1 = 3$. The points marked $t_1 = 2l, t_2 = 20l, t_3 = 40l$ denote the maximal best-case lapse times due to the limited length of the ramp (respectively to the speeds). The best-case scenario occurs only with probability $1/l$ (assuming a

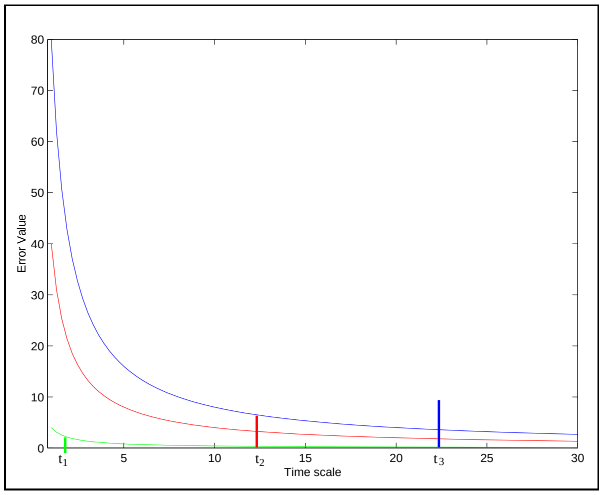


Figure 7: The two error functions values as as a function of time scale s. The red, green and blue graphs show the error of the temporal function for three different speeds $v = 0.5, 0.05, 0.025$ pixels/frame for $\sigma_1 = 3$. The points t_1, t_2, t_3 designate the best-case temporal window delay for the three velocities, respectively.

uniform distribution); therefore for an average initial sample at $l/2$, $t_1 = l, t_2 = 10l, t_3 = 20l$. In practice, the extent of l is never known, therefore determining t_i is hard to achieve. Instead, we use a temporal window over a region that is assumed to include t_i.

Figure 7 also suggests that assigning equal weights to all temporal scales does not lead to a minimal sum of errors in the case of small motions since the error is larger at the lower scales.

The 1D ramp example forms the basis for designing the weighting function for the errors of different scales. Since the weight function $W(u, v, s)$ should also reflect the width of the temporal window of computation of the flow estimation we redefine it to include a scaling parameter σ_w, $W(u, v, s, \sigma_w)$. W weighs the residual error contributions with respect to the total error; therefore, we choose to normalize its value so that $W = 1$ denotes full contribution and $W = 0$ denotes no contribution. The choice of the weighting function W should satisfy the following constraints:

- It should take on values in the range [0..1].

- The scaling parameter σ_w expresses the width of the temporal window over which flow estimates are integrated. Initially, we have no knowledge of the locally optimal time-scale for measurement of flow and, therefore, the weighting function weighs all residual errors equally (i.e, the temporal window is as wide as the number of frames involved in the estimation). But, as the estimates are iteratively refined, the weighting function should concentrate the weights on the most appropriate scales and ignore the remaining scales by narrowing the width of the window. Therefore, by first choosing a large σ_w, we ensure that W is uniformly 1.0 for all (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 8 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). We adopt the assumption of normal distribution of the sources of error, which naturally leads to Gaussian weighting functions. The following Gaussian function satisfies the above requirements

$$W(u, v, s, \sigma_w) = e^{-(s-\frac{n}{(\alpha||(u,v)||+1.0)})^2/2{\sigma_w}^2} \tag{10}$$

where $||(u, v)||$ is the magnitude of the current flow estimate at (x, y), and α is a constant. Notice that when $||(u, v)|| << 1.0$ the maximal weight occurs at the highest scale n, while higher values of $||(u, v)||$ lead to a maximal weight at lower scales; specifically the Gaussian is centered at $\frac{n}{\alpha||(u,v)||+1.0}$. The scale parameter σ_w determines the width of the Gaussian, and the constants α and 1.0 determine the maximal weight scale location. The application of the weighting function in the estimation is as follows: initially 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 [4] that constructs a pyramid of the spatially filtered and sub-sampled images and computes the flow initially at

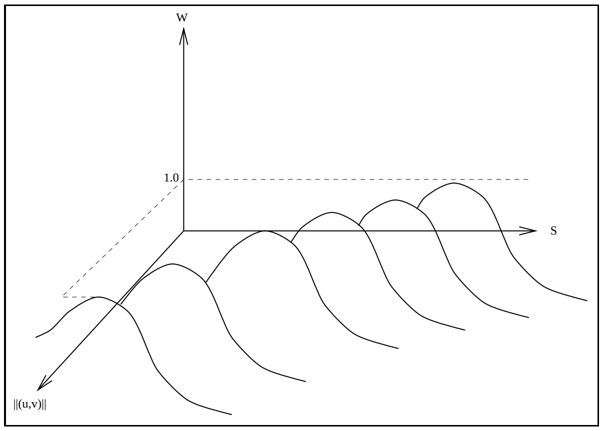


Figure 8: The weighting function as a function of s and flow magnitude $||(u, v)||$

the coarsest level and then propagates the results to finer levels. This allows dealing with large motions that may be present in addition to the small and very small motions. The computational aspects of the multi-scale model follow, generally, the approach proposed by Black and Anandan [5, 6]. Equation (8) can be rearranged as

$$E_D(u,v) = \sum_{(x,y)\in R} \rho\, (\sum_{s=1}^{n} W(u,v,s,\sigma_w)({I^s}_x u + {I^s}_y v + {I^s}_t), \sigma_e) \tag{11}$$

The minimization is carried out using a descent method, Simultaneous over-Relaxation (SOR). The minimization of $E_D(u, v)$ with respect to u and v is achieved using an iterative update equation, so at step $i + 1$

$${u_{x,y}}^{(i+1)} = {u_{x,y}}^{(i)} - \omega \frac{1}{T(u_{x,y})} \frac{\partial E_D}{\partial u_{x,y}} \tag{12}$$

where $0 < \omega < 2$ is an overrelaxation parameter which is used to overcorrect the estimate of ${u_{x,y}}^{(i+1)}$ at stage $i + 1$ (a similar treatment is used for v). The value of ω determines the rate of convergence. The term $T(u_{x,y})$ is an upper bound on the second partial derivative of E_D

$$T(u_{x,y}) \geq \frac{\partial^2 E_D}{\partial^2 u_{x,y}} \tag{13}$$

To achieve a globally optimal solution the robust error norm ρ is started with a large enough scale parameter σ_e to find a solution using the SOR technique. For each σ_e, the σ_w is started from a value that allows several scales to contribute to the solution; then it is gradually narrowed down to focus on the best scale. This process is iteratively repeated while decreasing σ_e and starting with the last estimate. The choice of a large enough σ_e guarantees convexity of the error function at the beginning of the process, which is followed by the use of the Graduated Non-Convexity method developed in [8]. The iterated decrease in σ_e reduces the influence of the outlier measurements and thereby refines the estimates while the reduction in σ_w improves the scale selection.

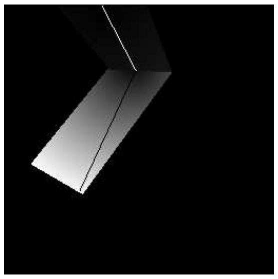
4 Experimental Results

4.1 A Synthetic Example

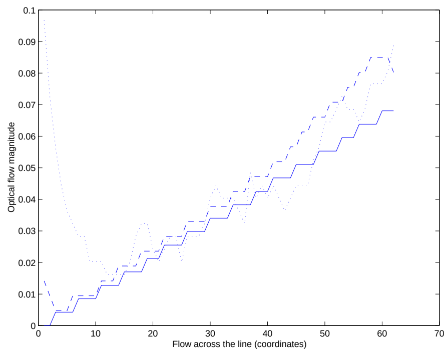
In the following figures we show the results of image flow computation when $\sigma_w = 5.0$ and is decreased at a rate of 0.6 for five iterations, and $\sigma_e = 18.0$ and is decreased at a rate of 0.4 for 5 iterations. The computation is performed over 12 temporal scales.

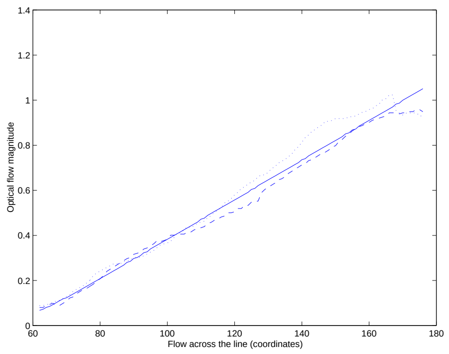
In order to compare the performance of single scale ($scale = 1$) and multi-scale flow estimation, we generated a sequence of images using a synthetic flow model where we have ground-truth data. Figure 9 (top) shows an image of a person during a walking activity. The synthetic sequence is generated by warping the image patch of part of the "thigh" and the whole "calf" foreward so that the thigh is rotating slowly while the "calf" is rotating faster around the knee. The sequence simulates a common motion in sequences of human motion. Figure 9 (middle row) shows quantitative comparison along a line on the "calf" (left to right) between the ground-truth 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 especially at very small flow magnitudes. Accurate recovery of the flow is actually limited by interpolation side effects in generating the synthetic motion.

Table 2 provides comparative results on the performance of single scale, multiple scale with equal



(a) (b) (c) (d)





Flow along the thigh

Flow along the calf

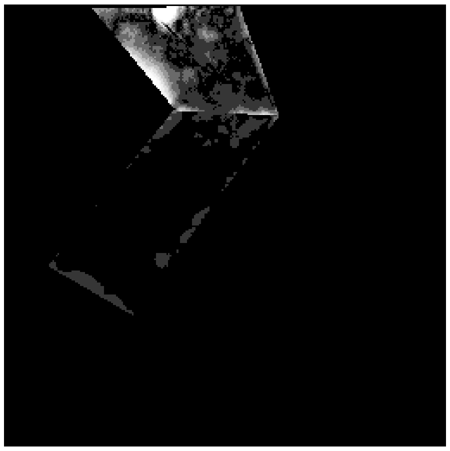
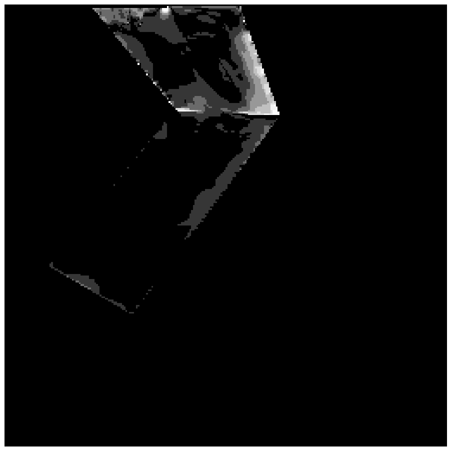



Figure 9: A synthetic motion example that compares flow magnitudes on a real image of a calf and thigh . The image (see (b)) was warped and the flow magnitudes along a line (see (c)) are shown as a solid line (see center row). The estimates of flow magnitudes using 1 scale and 12 scales over the same line are shown (center row, dotted and dashed lines, respectively). The lower row shows the magnitude of angular degree errors over the region for the 1 scale and 12 scale algorithms (left to right respectively). Notice that the slow-moving thigh is more error prone for the single scale algorithm.

Algorithm	Mean Angular Error	Standard Deviation
Single scale	2.8	3.13
Multi-scale W=1	4.95	7.81
Multi-scale Gaussian	2.5	2.91

Table 2: Comparative performance of algorithms by mean error and standard deviation of angular errors in the flow

Algorithm	$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
Single scale	28%	55%	68%	86%	99%
Multi-scale W=1	20%	41%	57%	72%	89%
Multi-scale Gaussian	41%	60%	70%	83%	97%

Table 3: Comparative performance of algorithms by percentage of points below $1°, 2°, 3°, 5°, 10°$ angular error

weights and multiple scales with Gaussian weighting with respect to the mean angular error and standard deviation of angular error as proposed in [3]. As in [3], the image velocities are represented as a 3D vector,

$$\vec{v} = \frac{1}{\sqrt{u^2 + v^2 + 1}}(u, v, 1)^T.$$

The error between the estimated flow, v_e and the real flow v_r is given by $arccos(v_r \cdot v_e)$. The multi-scale Gaussian algorithm is slightly better than the single scale and uniformly weighted multi-scale algorithms. It is illustrative to compare the percentage of points below specific angular errors. Table 3 shows the results for angular errors of $1, 2, 3, 5$ and 10. The multi-scale Gaussian results show improved accuracy.

4.2 A Real Sequence

In the following figures we show the results of image flow computation when $\sigma_w = 20.0$ and is decreased at a rate of 0.5 for five iterations, and $\sigma_e = 30.0$ and is decreased also at a rate of 0.5 for 5 iterations. The computation is performed over 16 temporal scales.

Figure 10 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

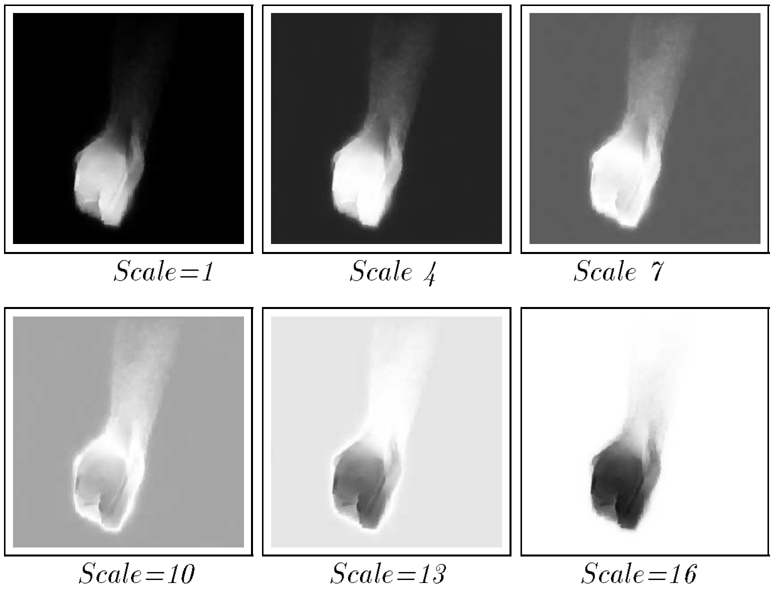


Figure 10: The weighting function W as computed at the scales 1,4,7,10,13 and 16 scales (top left to bottom right respectively) expressed as an intensity image.

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 11 shows the effect of the iterative refinement 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. This behavior is reversed when we consider the coarsest scale, $scale = 16$ (see Figure 12).

Figure 13 (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 4. Since the arm is *approximately* moving like a pendulum with the hand simultaneously rotating around the wrist (see Figure 18), the flow should increase slowly along the arm then jump considerably on the hand. This is clearly visible in the graphs. The dip in these graphs

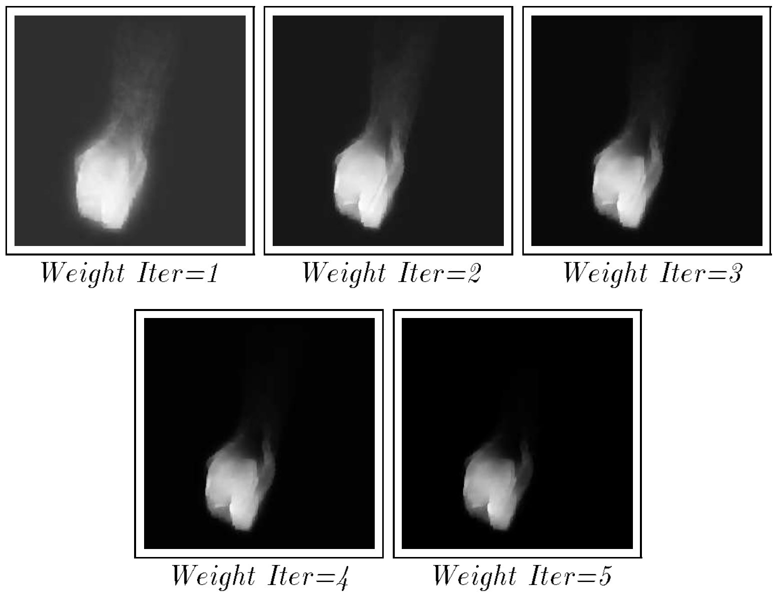


Figure 11: The weighting function W at scale 1 (finest scale) as evolved in five iterations

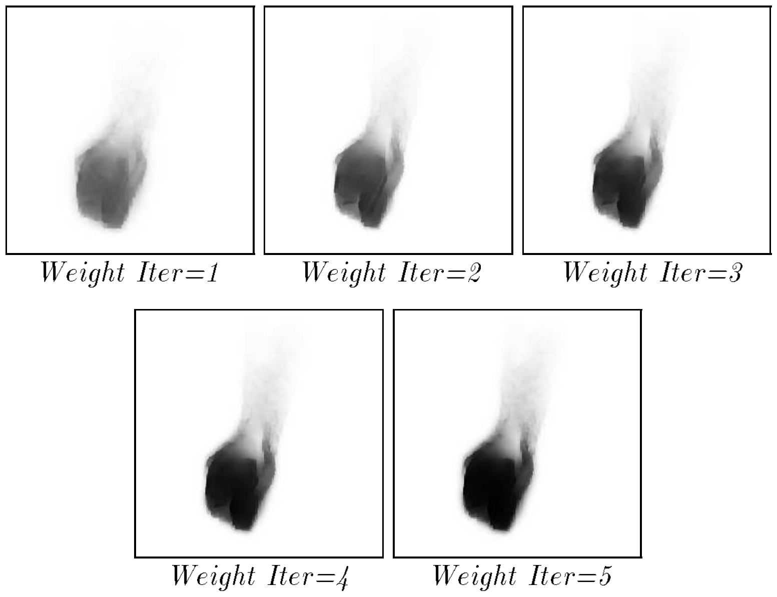


Figure 12: The weighting function W at scale 16 (coarsest scale) as evolved in five iterations

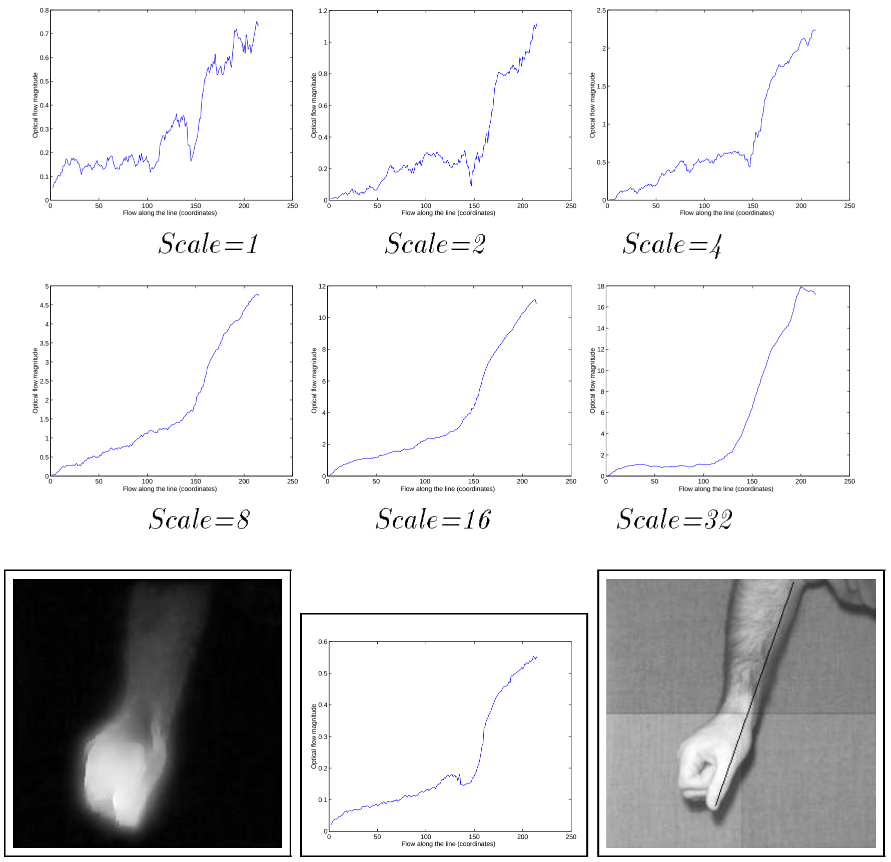


Figure 13: The flow magnitude along a line (bottom right) computed using a single scale ($s = 1, 2, 4, 8, 16$ and 32 scales; top and middle rows), the multi-scale flow magnitudes (bottom left), and the multi-scale flow magnitudes along the line (bottom center)

(occurring between 125-145) is a result of the intensity discontinuity of the four quadrants of the camera. Figure 13 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).

Figure 14 shows four images taken from a long sequence of a person moving his arm around while rotating his face (consecutive images are four frames apart). The magnitudes of the motions of the arm and face vary significantly. Also, the flow along the arm varies, beginning with no motion at the shoulder and increasing towards the hand. In Figures 15-16 the single-scale, multi-frame and multi-scale

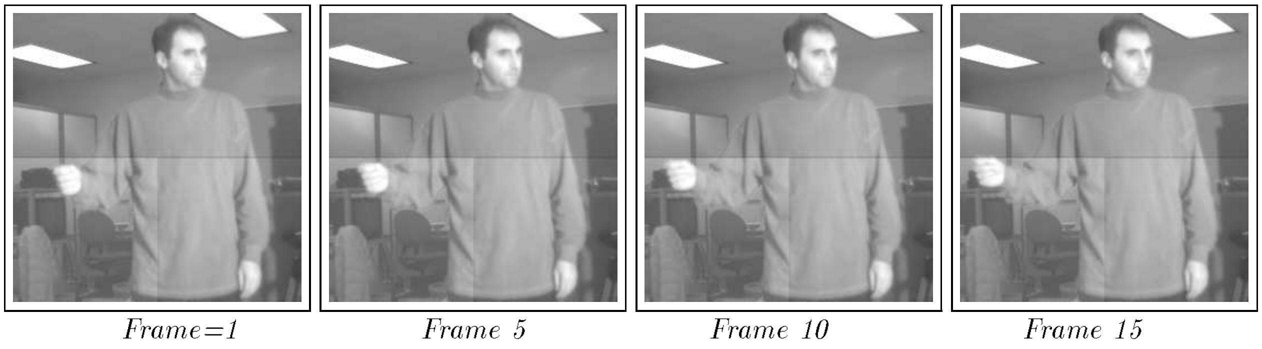


Figure 14: Images from a sequence of simultaneous arm and head motion

flow are shown (16 scales). Notice that the multi-frame flow computation is simply the case when W is set to one for all residuals (see top row in Figure 16). While these results are slightly better than the single-scale they still suffer from the inclusion of large residuals that the multi-scale algorithm recognizes and discards.

5 Estimation of Image Acceleration

The scale-generalized brightness constancy assumption given in Equation (2) 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 $(u(s), v(s))$. This image flow describes the instantaneous velocity between frame $s-1$ and s. Then the brightness constancy assumption at scale s becomes

$$I(x, y, t) = I(x + \sum_{j=1}^{s} u(j), y + \sum_{j=1}^{s} v(j), t + s) \tag{14}$$

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

$$u(s) = b_0 + b_1 s \tag{15}$$

$$v(s) = b_2 + b_3 s \tag{16}$$

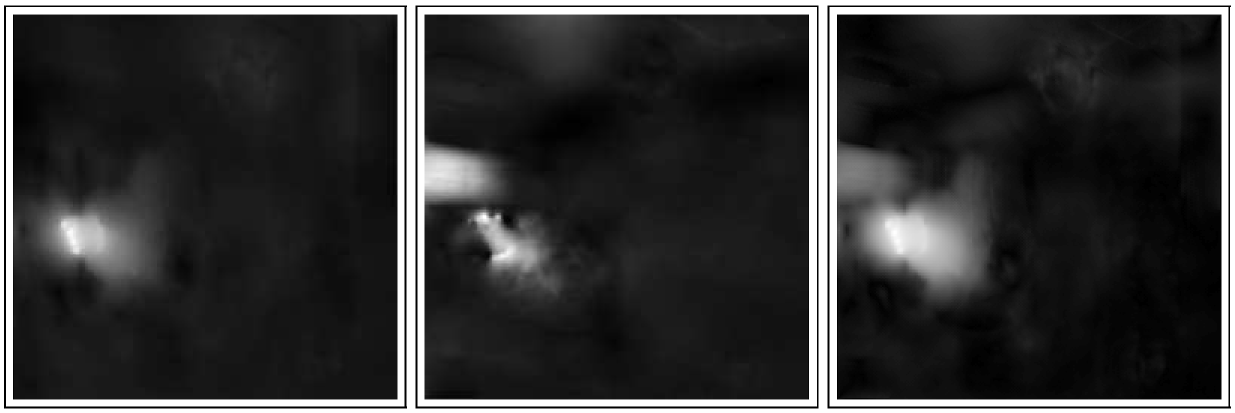


Figure 15: Horizontal, vertical and magnitude of flow from two frames (left to right). For horizontal and vertical flow, brighter value indicates greater motion leftward and upward, respectively.

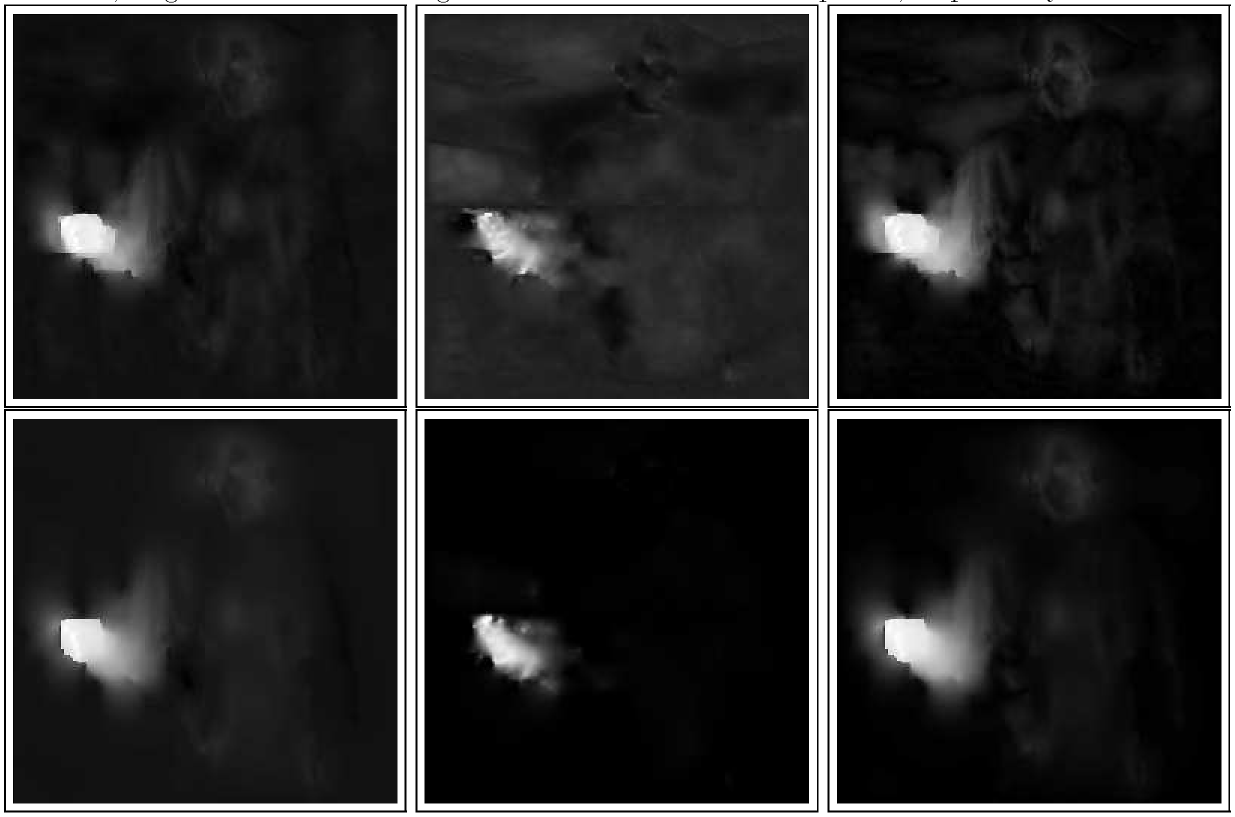


Figure 16: Top row, single-scale horizontal, vertical and magnitude of flow (left to right). Center row, multi-frame horizontal, vertical and magnitude of flow (left to right) (16 frames with the weight $W = 1$ for all points at all times. Bottom row, multi-scale ($s = 16$) horizontal, vertical and magnitude of flow (left to right). For horizontal and vertical flow, brighter value indicates greater motion leftward and upward, respectively.

where b_1 and b_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 (about 35% outliers) this model holds. This flow model leads to a brightness constancy assumption of the form

$$I(x,y,t) = I(x + \sum_{i=1}^{s}(b_0 + b_1 i), y + \sum_{i=1}^{s}(b_2 + b_3 i), t + s) \tag{17}$$

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

$$0 = I^s{}_x(b_0 + b_1 \frac{s+1}{2}) + I^s{}_y(b_2 + b_3 \frac{s+1}{2}) + I^s{}_t \tag{18}$$

The new scale-generalized error function is given by

$$E_D(u,v) = \sum_{s=1}^{\mathrm{n}} \sum_{(x,y)\in R} \rho(W(I^s{}_x(b_0 + b_1 \frac{s+1}{2}) + I^s{}_y(b_2 + b_3 \frac{s+1}{2})) + I^s{}_t), \sigma_e) \tag{19}$$

Figure 17 shows the dense flow and acceleration estimated for a book-falling sequence (see also Figure 19). 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.

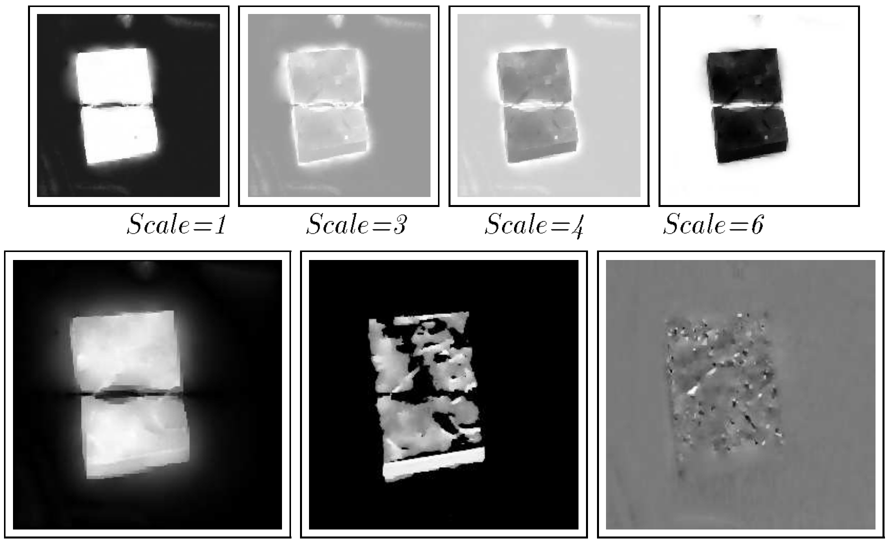


Figure 17: The weights (upper row) at scales 1, 3, 4 and 6, respectively (out of 6 scales), and the flow magnitude and vertical and horizontal accelerations (bottom row, left to right, respectively) for a falling book.

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 [7]). 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 (3) assumes constant flow over a small neighborhood around the point (x, y). Over larger neighborhoods, the image flow is given by low-order polynomials [1]. For example, affine motion is given by

$$U(x, y) = a_0 + a_1 x + a_2 y \tag{20}$$

$$V(x, y) = a_3 + a_4 x + a_5 y \tag{21}$$

where a_i's are constants and (U, V) is the instantaneous velocity vector. A more general model is the planar motion model [1] which is an approximation to the flow generated by a plane moving in 3-D under perspective projection. The planar motion model is given by

$$U(x,y) = a_0 + a_1 x + a_2 y + a_6 x^2 + a_7 xy \quad (22)$$

$$V(x,y) = a_3 + a_4 x + a_5 y + a_6 xy + a_7 y^2 \quad (23)$$

Equation (8) now becomes

$$E_D(U,V) = \sum_{s=1}^{n} \sum_{(x,y)\in A/P} \rho(W(U,V,s,\sigma_w)({I^s}_x U + {I^s}_y V + {I^s}_t), \sigma_e) \quad (24)$$

where A/P denotes the region in which the flow is assumed to be affine (A) or planar (P). The minimization of Equation (24) 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). This alternative reflects the fact that the region is treated as a single object that is subject to rigid motion; therefore its motion is completely captured by the parameters $\bar{a}$. 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 [6].

In the examples in this section we adopt the weighting of parametric models. Recall that the parameters of the affine and planar models capture several aspects of the region's motion (see [7]). 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 (10) so that it becomes:

$$W(u, v, s, \sigma_w) = e^{-(s - \frac{n}{(\alpha||(a_0, a_3)||+1.0)})^2/2\sigma_w{}^2} \tag{25}$$

Figure 18 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 motion throughout the sequence similar to [7] (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.

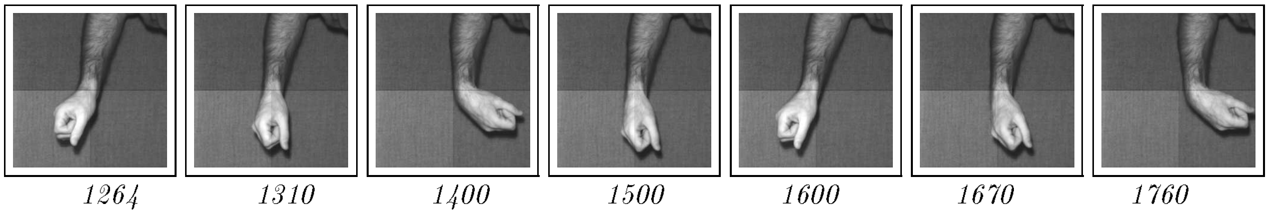
Parameterized flow models can also be extended to include acceleration. The extension of the affine model requires that the motion parameters across scales be dependent on the scale so that a_i becomes $a_i(s)$. Assuming a constant acceleration for these parameters, the models now become

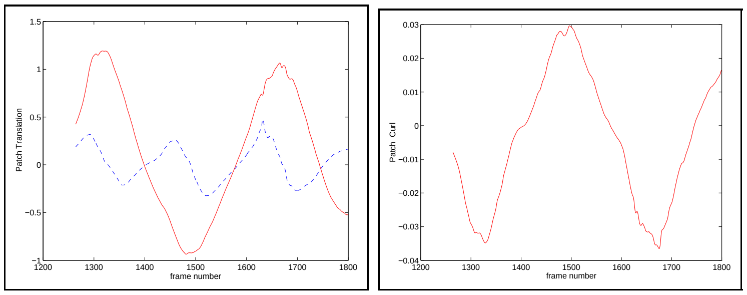
$$U(x, y) = (a_0 + a_0{}'s) + (a_1 + a_1's)x + (a_2 + a_2's)y \tag{26}$$

$$V(x, y) = (a_3 + a_3{}'s) + (a_4 + a_4's)x + (a_5 + a_5's)y \tag{27}$$

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 19 describes an experiment in which the acceleration of a falling book is estimated from an image





Flow results without acceleration model, a_3 and a_0
(left, solid and dashed lines, respectively) and curl (right).

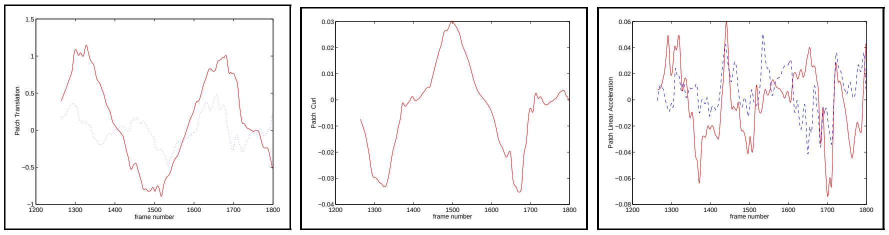


Figure 18: Seven images of a long sequence of the arm in motion (top). Flow results of teh hand tracking without acceleration model, a_3 and a_0 (left, solid and dashed lines, respectively) and curl (center row left to right). Flow results with an acceleration model, a_3 and a_0 (left, solid and dashed lines, respectively), curl (bottom left to right) and a_3' and a_0' acceleration (bottom right).

sequence.[§] The computation is done over 16 scales. 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 19 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 in the image motion model is valuable in predicting the real motion for a significant amount of time.

Figure 18 shows the acceleration of the hand in addition to the flows for long image sequences. The

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

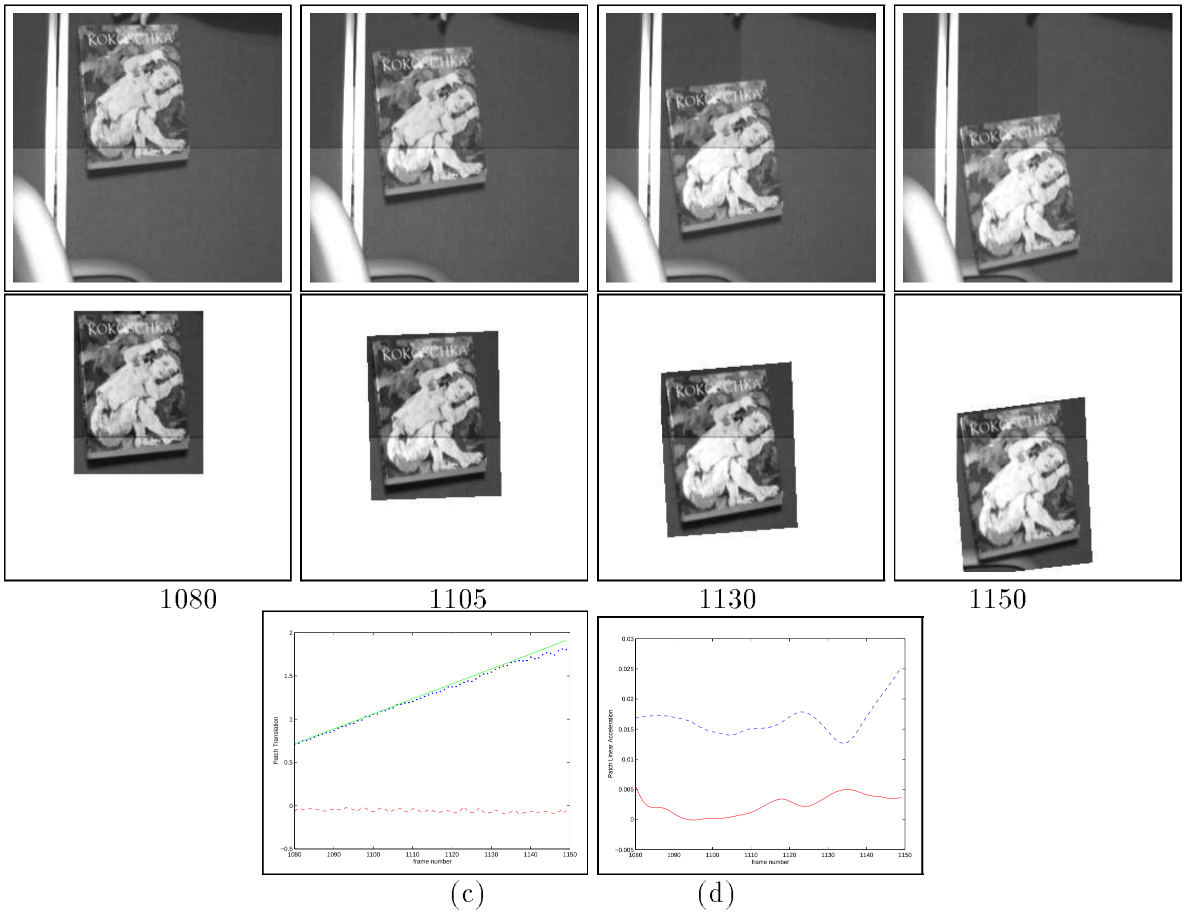


Figure 19: Four frames of a falling book (a)-(b) from an 80 frame sequence, the vertical and horizontal velocities (dashed and dotted lines, respectively (c)), and the predicted vertical velocity from the starting velocity and average acceleration, a_3', in the first 10 frames (solid line, (c)) and the vertical and horizontal accelerations a_3', a_0' (dashed and solid lines respectively, (d))

acceleration of the hand does not satisfy the constant acceleration model, thus the estimates are not linear.

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 typically 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 computes temporal weights that control the contribution of the spatio-temporal derivatives at each time instance and thereby improve the accuracy of the instantaneous motion estimates.

In this paper we developed a new multi-temporal method 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 thank Michael Black for providing the code for optical flow estimation from two images. We also thank the anonymous reviewers for their thorough reviews that enhanced the presentation in this paper.

References

[1] Adiv G. *Determining three-dimensional motion and structure from optical flow generated by several moving objects.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 7(4), 1985, 384-401.

[2] R. Battiti, E. Amaldi and C. Koch. *Computing Optical Flow Across Multiple Scales: An Adaptive Coarse-to-Fine Strategy.* International Journal of Computer Vision, (6)2, 1991, 133-145.

[3] S.S. Beauchemin and J.L. Barron. *The Computation of Optical Flow.* ACM Computing Surveys, (27)3, 1995, 433-467.

[4] 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, Springer-Verlag, 1992, 237-252.

[5] M.J. Black and P. Anandan. *A Frame-work for Robust Estimation of Optical Flow.* International Conference on Computer Vision, 1993, 231-236.

[6] 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), 1996, 75-104.

[7] 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), 1997, 23-48.

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

[9] D.J. Fleet and A.D. Jepson. *Computation of Component Image Velocity from Local Phase Information.* International Journal of Computer Vision, 3(4), 1990, 77-104.

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

[11] D.J. Heeger. *Optical Flow Using Spatio-temporal Filters.* International Journal of Computer Vision, 1, 1988, 279-302.

[12] J. Kearney. W.B. Thompson and D.L. Boley. *Optical Flow Estimation: An Error Analysis of Gradient-Based Methods with Local Optimization.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(2), 1987, 229-244.

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

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

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

[16] J. Weber and J. Malik, *Robust Computation of Optical Flow in a Multi-Scale Differenial Framework.* International Jounral of Computer Vision, 14(1), 1995, 67-81.