Using 3D Spline Differentiation to Compute Quantitative Optical Flow

John Leonard Barron[1]
[1]Middlesex College
Dept. of Computer Science
University of Western Ontario
London, Ontario, Canada, N6A 5B7
barron@csd.uwo.ca

Marc Daniel[2] - Jean-Luc Mari[2]
[2]Laboratoire des Sciences de l'Information et des Systèmes
Ecole Supérieure d'Ingénieurs de Luminy
Case Postale 925
13288 Marseille, Cedex 9, France
{*mdaniel,jlmari*}*@esil.univ-mrs.fr*

Abstract

We show that differentiation via fitting B-splines to the spatio-temporal intensity data comprising an image sequence provides at least the same and usually better 2D Lucas and Kanade optical flow than that computed via Simoncelli's balanced/matched filters.

Keywords: B-Splines, Filters, Differentiation, 2D Optical Flow, Quantitative Error Analysis

1 Introduction

This paper quantitatively compares 2D optical flow computed using derivatives measured with the same spatio-temporal support via Simoncelli's matched/balanced filters [16] and via B-spline differentiation. As Unser [21] points out, splines are a natural way to convert discrete data into continuous data and splines have already been used in optical flow work. For example, the work by Goshtasby [10] applied surface fitting to discrete displacement estimates based on feature correspondences to obtain smooth displacement fields. Bab-Hadiashar et al. [1] proposed an optical flow computation that involves fitting 2D thin-plate spline patches to image intensity data on a rectangular grid via an LU decomposition. This method is different from those using a spline to smooth the optical flow field [18]. Szeliski and Couchlan [19] represent the optical flow, (u, v) using 2D splines controlled by a smaller number of displacement estimates, $(\hat{u}, \hat{v})$, which lie on a coarser 'control point' grid. Our idea is to use B-splines directly to obtain a smooth spatio-temporal grayvalue model of the spatio-temporal grayvalue intensities that make up an image sequence and use this model to compute analytic 1^{st} order derivatives in x, y, and t. We believe our use of a 3-variate spline with a uniform knot vector is unique in this context. Modeling with B-splines offers an extra important flexibility because of the different smoothing parameters. An introduction to B-spline theory is proposed in [6] and some background on their use can be found in [7].

We use the computed flow from Lucas and Kanade's method to compute the flow for synthetic images sequences with known groundtruth for derivatives computed both ways: we hypothesize the error in the flow indicates which type of derivatives are more numerically accurate. As such, it shouldn't matter very much which optical flow method we use. We choose Lucas and Kanade because it is computationally efficient (versus, say, Horn and Schunck's regularization method [11] which is significantly slower), provides accurate flow and because we already have a good implementation of this algorithm.

2 Lucas and Kanade Optical Flow Review

The basis of all differential optical flow methods is the motion constraint equation,

$$I_x v_x + I_y v_y = -I_t, \tag{1}$$

where I_x, I_y and I_t are the spatio-temporal image intensity derivatives at a pixel and (v_x, v_y) are the x and y components of the 2D image motion (image velocity) at that pixel. Equation (1) is one equation in two unknowns. Any velocity that satisfies this equation is potentially the correct velocity. If we assume the motion is constant in some neighbourhood

than two (or more) sets of derivatives can yield a non-singular linear (least squares) system of equations that yield values for (v_x, v_y).

The velocity with the smallest magnitude on the motion constraint equation is called the **normal** velocity as it is one of the endpoints on the shortest line from that point to the origin (this line is perpendicular [normal] to the motion constraint line) and is given by $\vec{v}_n = v_n \hat{n}$, where

$$v_n = \frac{-I_t}{||\nabla I||_2} \quad \text{and} \quad \hat{n} = \frac{(I_x, I_y)}{||\nabla I||_2} \tag{2}$$

are the normal velocity magnitude and the normal velocity direction and $\nabla I = (I_x, I_y)$ is the spatial intensity gradient.

We implemented a weighted least-squares fit of the local first-order constraints (1) based on Lucas and Kanade [12, 3] to a constant model for $\mathbf{v}$ in each small spatial neighbourhood Ω by minimizing

$$\sum_{(x,y)\in\Omega} \left[W^2(\nabla I \cdot \mathbf{v} - I_t)^2\right], \tag{3}$$

where Ω is some local neighbourhood and W denotes the diagonal weight matrix whose diagonal elements are the magnitudes of the spatial gradients, $||\nabla I||_2$. The solution to equation (3) is given by

$$M^T W^2 M \mathbf{v} \;=\; M^T W^2 \mathbf{b}\,, \tag{4}$$

where, for n points $(x, y) \in \Omega$ at a single time t,

$$\begin{aligned} M_i &= [I_{xi} \;\; I_{yi}], i \in [1, n], \\ W &= \text{diag}[p_1, p_2, ..., p_n], \\ \mathbf{b} &= -(I_{t1} \;\; I_{t2} \;\; ... \;\; I_{tn})^T. \end{aligned} \tag{5}$$

The solution to (4) is $\mathbf{v} = [M^T W^2 M]^{-1} M^T W^2 \mathbf{b}$, which is solved in closed form when $M^T W^2 M$ is non-singular, since it is a 2×2 matrix:

$$M^T W^2 M \;=\; \begin{bmatrix} \sum W_i^2 I_{xi}^2 & \sum W_i^2 I_{xi} I_{yi} \\ \sum W_i^2 I_{yi} s_{xi} & \sum W_i^2 I_{yi}^2 \end{bmatrix}, \tag{6}$$

where all sums are taken over points in the local spatial 5×5 neighbourhood Ω. The solution to this system of equations is a least squares fit of a single full velocity to a local 5×5 neighbourhood of spatio-temporal intensity derivatives, i.e. a constant local velocity model is assumed for each neighbourhood. With respect to the motion constraint equation in equation (1) this can be viewed as a least squares computation of the intersection point of all the motion constraint lines arising from the normal velocities in the neighbourhood.

We used eigenvalue/eigenvector decomposition to threshold the computed flows. Eigenvalue ($\lambda_0 \leq \lambda_1$) and corresponding eigenvector ($\hat{e}_0$ and $\hat{e}_1$) decomposition of the symmetric matrix $M^T W^2 M$ yields least squares full image velocity, if both $\lambda_0, \lambda_1 > \tau$, or least squares normal image velocity, $\mathbf{v}_n = \mathbf{v} \cdot \hat{e}_1$, if $\lambda_1 > \tau$ but $\lambda_0 \leq \tau$ [3]. We use $\tau = 1.0$ for the full image velocity results reported in this paper.

3 Simoncelli Differentiation

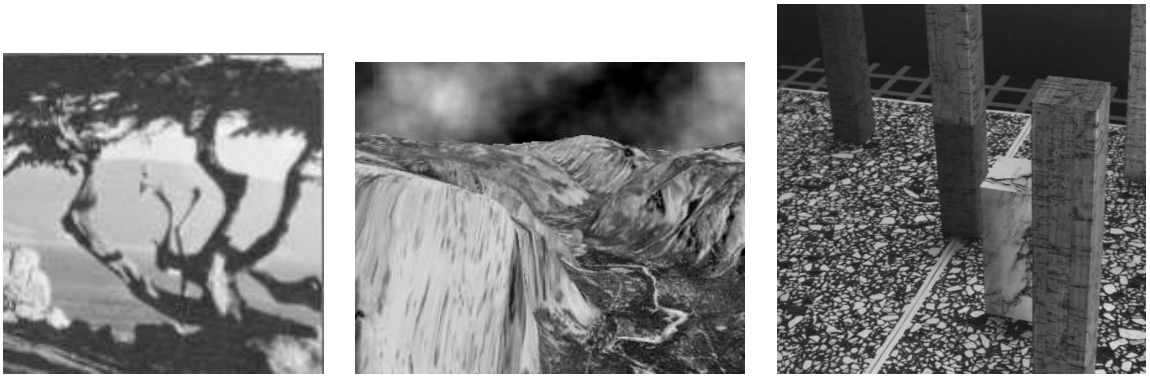
Differentiation via convolution was done using Simoncelli's matched/balanced filters [16]. Simoncelli designed filters that required the derivative kernels to be good approximations to the derivatives of the prefilter, are linear, have zero phase and are separable. Let $\vec{p}$ and $\vec{d}$ be N-vectors containing the kernels for a prefilter and its derivative. If $P(\vec{\omega})$ and $D(\vec{\omega})$ are the FTs of $\vec{p}$ and $\vec{d}$ respectively, then a weighted least squares minimization in the frequency domain of:

$$\int W(\omega) \left[-i\omega P(\omega) - D(\omega)\right]^2 d\omega, \tag{7}$$

where $W(\omega) = 1/|\omega|^{0.5}$ is a weighting function, will yield the N tap kernels for the prefilter and its derivative. A discrete least squares minimization of equation (7) requires us to minimize:

$$(WF' | WF) \begin{pmatrix} \vec{p} \\ \vec{d} \end{pmatrix}, \tag{8}$$

where F is a matrix containing the first N Fourier basis functions, F' is the same matrix multiplied by frequency ω and W is a weighting matrix (as above). This minimization is performed by finding the minimum eigenvalue and corresponding eigenvector of $M^T M$. We use $N = 5$ which gives a low pass filtering (blurring), p_5, and a high pass filtering (differentiation), d_5, as specified in Table 1. Matched filters allow comparisons between the signal and its derivatives as the high pass filter is simply the derivative of the low pass filter and should yield more accurate derivative values. Using these two masks I_x is computed by applying p_5 in the t dimension, then p_5 to those results in the y dimension and finally d_5 to those results in the x dimension. I_y and I_t are computed in a similar manner. Before performing this filtering we use a simple averaging filter $[\frac{1}{4}, \frac{1}{2}, \frac{1}{4}]$ to slightly blur the images. Simoncelli claims that because both of his filters were derived from the same principles more accurate derivatives result and he demonstrated this on the Yosemite Fly-Through sequence [16].



(a) (b) (c)

Figure 1: Synthetic image data: (a) The middle translating tree image (the middle diverging image is very similar and not shown), (b) the middle Yosemite-Fly Through image and (c) the middle marble image.

n	p_5	d_5
0	0.036	-0.108
1	0.249	-0.283
2	0.431	0.0
3	0.249	0.283
4	0.036	0.108

Table 1: Simoncelli's 5-point Matched/Balanced Kernels

4 Spline Differentiation

We use B-splines to parameterize the spatio-temporal grayvalue data. Then the spatio-temporal grayvalue derivatives, I_x, I_y and I_t can be computed by analytically differentiating the spline. A B-spline curve of order m (degree $m-1$) can be defined as

$$P(t) = \sum_{k=0}^{L} P_k N_{k,m}(t), \tag{9}$$

where a knot vector $T = (t_0, t_1, , ..., t_{L+m})$ must be specified, there must be $L+1$ control points, P_k (pixel intensity values in our case) and the order m of the B-spline functions. Typically, cubic B-splines ($m = 4$) are the most common because they produce a C^2 curve with the lowest number of calculations. The B-spline function, $N_{k,m}$ is defined recursively as [6]:

$$N_{k,m}(t) = \left(\frac{t - t_k}{t_{k+m-1} - t_k}\right) N_{k,m-1}(t) + \left(\frac{t_{k+m} - t}{t_{k+m} - t_{k+1}}\right) N_{k+1,m-1}(t). \tag{10}$$

Since this is a recursive definition that defines the m^{th} order function in terms of 2 B-spline $(m-1)^{th}$ order functions we need to define the 1^{th} order function:

$$N_{k,1}(t) = \begin{cases} 1 & \text{if } t_k < t \leq t_{k+1} \\ 0 & \text{otherwise} \end{cases}. \tag{11}$$

We can differentiate Equation (10) with respect to t to obtain:

$$N'_{k,m}(t) = (m-1)\left(\frac{N_{k,m-1}(t)}{t_{k+m-1} - t_k}\right) + \left(\frac{N_{k+1,m-1}(t)}{t_{k+m} - t_{k+1}}\right). \tag{12}$$

Equation (12) provides the explicit derivation of expression (9). Reorganizing the result leads to a most classical expression of the derivative [7], but also entails more computations. We can smooth an image sequence segment as a 3D spline of order m_u, m_v, m_s by computing $C(u, v, s)$ as:

$$\sum_{i=0}^{n_u} \sum_{j=0}^{n_v} \sum_{k=0}^{n_s} N_{i,m_u}(u) N_{j,m_v}(v) N_{k,m_s}(s) I_{i,j,k}, \tag{13}$$

where $I_{i,j,k}$ is the image intensity at pixel (i, j) at frame k. n_u, n_v and n_s are the dimensions of image sequence in the spatio-temporal dimensions. Using such a smoothing, intensity derivatives, $I_x(i, j, k)$, $I_y(i, j, k)$ and $I_t(i, j, k)$, reduce to:

$$I_x(i, j, k) = C_u(u, v, s) \times \frac{du}{dx} \tag{14}$$

$$= \frac{du}{dx} \times \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} \sum_{k=0}^{n_s} N'_{i,m_u}(u) N_{j,m_v}(v) N_{k,m_s}(s) I_{i,j,k},$$

$$I_y(i,j,k) = C_v(u,v,s) \times \frac{dv}{dy} \tag{15}$$

$$= \frac{dv}{dy} \times \sum_{i=0}^{n_u}\sum_{j=0}^{n_v}\sum_{k=0}^{n_s} N_{i,m_u}(u)N'_{j,m_v}(v)N_{k,m_s}(s)I_{i,j,k}$$

and

$$I_t(i,j,k) = C_s(u,v,s) \times \frac{ds}{dt} \tag{16}$$

$$= \frac{ds}{dt} \times \sum_{i=0}^{n_u}\sum_{j=0}^{n_v}\sum_{k=0}^{n_s} N_{i,m_u}(u)N_{j,m_v}(v)N'_{k,m_s}(s)I_{i,j,k}.$$

2^{nd} order derivatives can be computed in a similar way if required. The image sequence parameterization depends on the 3D data distribution and the knot vectors. In our case, the data are regularly spaced in x, y and t. We also use a uniform knot vector. When a set of $n+1$ points is smoothed with a B-spline of order k, the knot vector must exactly have $n+k+1$ knots, $t_0, ..., t_{n+k}$. The first knots are starting knots and the last ones are ending knots. The definition interval of the corresponding B-spline model is $[t_{k-1}, t_{n+1}]$ [6]. We always use uniform knot vectors in our calculations, i.e. the step between two consecutive knots is always the same (the easiest step to use is 1). Using such a sequence (as opposed to a classical clamped knot vector) does not constrain the model to reach the extreme control points, which, in any case, would not be very useful as the latter are the inaccurate borders of an image. Moreover, having uniform knot vectors to smooth our regular lattice of points implies that there exist affine maps between respective parameters x, y, t and u, v, s. The first computed x value is x_{min} while the last one is x_{max}. With our parameterization, in a 512×512 image, x_{min} is not 0 and the value x_{max} is not 511. However, both values are within the range $[0, 511]$. From applying the affine map, it is simple to compute the first and last parametric values corresponding to integer values of x, y and t. Thus, we obtain the clipping window, within which the derivatives can be computed reliably with respect to x, y and t. The smaller the B-spline orders, the larger the windows and the number of frames, but the smaller the smoothing effect. One can note that determining a xyt clipping window is not a problem as our method reduces the number of pixels (at spatio-temporal borders) where we can't compute good derivatives. If the reduction in the number of the computed frames becomes a drawback, three different solutions can be proposed. First, a low B-spline order in s can be fixed (the minimal value being 2, suppressing the smoothing in time). The second is to consider a different parameterization allowing one to reach the extreme frames. The last solution is to use periodic B-splines when the frames correspond to the sampling of a periodic motion [20].

Computing a smoothed image is always a compromise between smoothness, accuracy and computational time. In order to obtain better results with low degree splines, we first pre-smooth the image sequence by parameterizing it with a spatio-temporal spline. The order of the B-splines used for smoothing was fixed at $m_{u,v,s} = 4$. Then we resample the image sequence from this parameterization and again fit a B-spline to this re-sampled data. This method is believed to yield more accurate derivatives even though if we still plan additional investigations.

Another advantage of computing derivatives using a spatio-temporal B-spline is that multiple sets of derivatives can be computed, yielding multiple optical flow fields. Using Simoncelli filters on the same number of images in a sequence (7 in our case) yields only 1 optical flow field where as using spline differentiation may provide more. We analyze the accuracy/goodness of these flows below.

5 Experimental Technique

We use 4 synthetic image sequences with known truth, the **Translating Tree** and the **Diverging Tree** sequences made by David Fleet [8], the **Yosemite** Fly-Through sequence made by Lynn Quan at SRI and the **Marble** sequence made by Otte and Nagel [14]. Figure 1 shows the middle image of each sequence. Due to space limitations, we do not show the correct flow fields in this paper but refer the interested reader to papers by Barron et al. [3] and Otte and Nagel [14].

We measure average magnitude and angle error (plus standard deviations) for the computed flows. Average magnitude error is computed as:

$$\phi_{2Dmag} = \frac{\sum_i^N \mid |\vec{v}_{c_i}| - |\vec{v}_{e_i}| \mid}{|\vec{v}_{c_i}|} \times 100\%. \tag{17}$$

Average angle error is computed as:

$$\phi_{2D\theta} = \arccos\left(\frac{\vec{v}_{c_i} \cdot \vec{v}_{e_i}}{||\vec{v}_{c_i}||_2||\vec{v}_{e_i}||_2}\right). \tag{18}$$

We also use the Fleet and Jepson 3D angular measure [8, 3]. In this case, velocity is viewed as displacement per time unit as in $\vec{v} = (u, v)$ pixels/frame, or as a space-time direction vector $(u, v, 1)$ in units of (pixel, pixel, frame). Then error is measured as angular deviation from the correct space-time orientation. Let 3D estimated velocity be expressed as the normalized vector $\hat{v}_e = \frac{(u_e, v_e, 1)}{\sqrt{u_e^2+v_e^2+1}}$ while the 3D

correct velocity is expressed as the normalized vector $\hat{v}_c = \frac{(u_c, v_c, 1)}{\sqrt{u_c^2+v_c^2+1}}$. The angular error is then computed as:

$$\psi_{3D\theta} = \arccos(\hat{v}_e \cdot \hat{v}_e). \quad (19)$$

Lastly, we present the density of the flow as a percentage out of 100. This is because we threshold Lucas and Kanade flow fields on the smallest eigenvalue of the least squares integration matrix using $\tau = 1.0$.

6 Experimental Results

In this section we present quantitative results of the computed flows for our 4 synthetic sequences. Tables 2 to 5 show our quantitative spline results for the **Translating Tree**, **Diverging Tree**, **Yosemite** Fly-Through and **Marble** sequences while Table 6 show our quantitative Simoncelli results for the middle images of those 4 sequences. Figures 2 to 5 show 3 representative flow for each of the 4 sequences: the flows at the middle image when the spatio-temporal spline order is $m_{u,v,s} = 4$ and when the spatio-temporal order is $m_{u,v,s} = 6$ and when Simoncelli's filters are used for $\tau = 1.0$. The results in the Tables show that as the spatial orders, m_u and m_v ($m_u = m_v$ always here) and the temporal order, m_s, are increased, the flows usually become more accurate. Since the computed flows are compared to the correct flow for the middle image, non-middle flows are only difference measures with respect to the middle image flow; nevertheless the time-varying flow varies smoothly and slowly and we can see that they often are in good agreement with the correct middle image flow. We perform these calculations to show that the multiple flow fields often available with spline differentiation are usually quite good (this relies on the assumption that the flow fields vary smoothly and slowly over time). In most cases, the Simoncelli-filters flows are often significantly inferior to the best spline flows for the error when the image number matches the correct flow number. We have also run these same tests with no spline presmoothing; in all cases, the results were inferior and are not reported here.

7 Conclusions

We have shown that, for 4 realistic synthetic images sequence with known groundtruth, that we can compute more accurate intensity derivatives, and as a result, more accurate optical flow by using derivatives computing via B-splines versus convolution with filters (known to be good). As well, we can use splines to compute more dense derivative datasets from 7 images. The Simoncelli filters we used only allow 1 flow field computation for each 7 images. A good question (not addressed in this paper) is one concerning noise and smoothing. B-splines provide exact derivatives on the interpolating function (in fact, the smoothing function). These derivatives are exact for the model (parameterization) that models the data. This is an important advantage combined with the fact that the different parameters provide a high degree of flexibility for smoothing (for noise reduction). One can choose to increase smoothing or to extend the interpolation parameterization. We are currently investigating this matter. Other future work includes designing a measure of "goodness of spline fit" to the data to allow calculation of reliability measures for the derivatives that can then be used to improve subsequent optical flow computation [17]. We could also use a better optical flow algorithm, such as Brox et al.'s regularization algorithm [5][1] with our spline differentiation; an implementation of this is now underway. We are also in the process of using spline differentiation to compute 3D optical flow on gated MRI datasets [4].

Acknowledgements

This work was supported in part by NSERC (Natural Sciences and Engineering Research Council of Canada) through a Discovery grant.

References

[1] A. Bab-Hadiashar, D. Suter and R. Jarvis (1995), "Optic Flow Computation Using Interpolating Thin-Plate Splines", Asian Computer Vision Conference (ACCV1995), pp452-456.

[2] J.L. Barron (2003), "The Integration of Optical Flow into Tinatool", Dept. of Computer Science, The Univ. of Western Ontario, TR601 (report for OSMIA (Open Source Medical Image Analysis, a EU 5^{th} Framework Programme project).

[3] J.L.Barron, D.J. Fleet and S.S. Beauchemin (1994), "Performance of optical flow techniques", *Intl. Journal of Computer Vision*, 12(1), IJCV1994, pp. 43–77.

[1] Won the best paper award at ECCV2004.

Image	$m_{u,v}/m_s$	ϕ_{2DMag}	$\phi_{2D\theta}$	$\phi_{3D\theta}$	Density
8	4/4	$21.96\% \pm 20.23\%$	$6.59° \pm 9.50°$	$7.57° \pm 8.33°$	56.01%
9	4/4	$6.18\% \pm 7.18\%$	$1.69° \pm 2.85°$	$2.18° \pm 2.99°$	54.78%
10	4/4	$6.38\% \pm 7.14\%$	$1.73° \pm 2.73°$	$2.24° \pm 2.90°$	54.81%
11	4/4	$21.20\% \pm 20.85\%$	$6.26° \pm 8.71°$	$7.28° \pm 7.86°$	55.82%
7	5/4	$39.24\% \pm 28.48\%$	$12.98° \pm 17.86°$	$17.09° \pm 14.82°$	68.26%
8	5/4	$21.48\% \pm 19.25\%$	$6.39° \pm 9.18°$	$7.34° \pm 8.02°$	54.67%
9	5/4	$5.66\% \pm 6.42\%$	$1.56° \pm 2.50°$	$1.98° \pm 2.58°$	53.95%
10	5/4	$5.88\% \pm 6.51\%$	$1.61° \pm 2.55°$	$2.06° \pm 2.63°$	53.90%
11	5/4	$20.72\% \pm 20.13\%$	$6.05° \pm 8.43°$	$7.03° \pm 7.59°$	54.80%
7	6/4	$35.62\% \pm 26.79\%$	$11.27° \pm 15.49°$	$14.99° \pm 13.33°$	65.34%
8	6/4	$21.20\% \pm 18.58\%$	$6.29° \pm 9.02°$	$7.20° \pm 7.87°$	53.76%
9	6/4	$5.19\% \pm 5.75\%$	$1.44° \pm 2.28°$	$1.81° \pm 2.27°$	52.90%
10	6/4	$5.42\% \pm 5.92\%$	$1.51° \pm 2.37°$	$1.90° \pm 2.38°$	53.10%
11	6/4	$20.45\% \pm 19.60\%$	$5.93° \pm 8.32°$	$6.89° \pm 7.46°$	53.89%
8	4/5	$22.66\% \pm 21.34\%$	$7.09° \pm 10.12°$	$8.07° \pm 8.91°$	56.10%
9	4/5	$4.60\% \pm 5.64\%$	$1.29° \pm 1.95°$	$1.61° \pm 2.11°$	54.11%
10	4/5	$4.67\% \pm 5.67\%$	$1.31° \pm 2.03°$	$1.64° \pm 2.19°$	54.08%
8	5/5	$22.05\% \pm 20.21\%$	$6.86° \pm 9.79°$	$7.78° \pm 8.58°$	54.83%
9	5/5	$4.19\% \pm 4.99\%$	$1.19° \pm 1.79°$	$1.47° \pm 1.84°$	53.23%
10	5/5	$4.29\% \pm 5.10\%$	$1.22° \pm 1.89°$	$1.50° \pm 1.96°$	53.27%
8	6/5	$21.68\% \pm 19.52\%$	$6.69° \pm 9.53°$	$7.57° \pm 8.34°$	53.89%
9	6/5	$3.86\% \pm 4.44\%$	$1.11° \pm 1.60°$	$1.35° \pm 1.59°$	52.49%
10	6/5	$3.95\% \pm 4.56\%$	$1.14° \pm 1.76°$	$1.38° \pm 1.75°$	52.64%
9	4/6	$4.35\% \pm 5.19\%$	$1.26° \pm 1.76°$	$1.52° \pm 1.83°$	53.61%
8	5/6	$22.11\% \pm 20.74\%$	$7.02° \pm 10.12°$	$7.96° \pm 8.91°$	54.93%
9	5/6	$4.01\% \pm 4.59\%$	$1.16° \pm 1.57°$	$1.40° \pm 1.57°$	52.86%
10	5/6	$3.96\% \pm 4.64\%$	$1.15° \pm 1.67°$	$1.39° \pm 1.67°$	52.96%
9	6/6	$3.76\% \pm 4.15\%$	$1.10° \pm 1.45°$	$1.31° \pm 1.40°$	52.23%
10	6/6	$3.69\% \pm 4.21\%$	$1.08° \pm 1.57°$	$1.29° \pm 1.52°$	52.27%

Table 2: Error statistics for the **Translating Tree** sequence for images 7-11. For images 7 and 8 as well as 10 and 11 the error measures are difference measures from the correct flow field, which is known for image 9 only.

[4] John Leonard Barron, Jean-Luc Mari and Marc Daniel (2004), "Towards Parameterising Gated MRI Cardiac Sequences to Compute Accurate Spatio-Temporal Derivatives for 3D Optical Flow", Groupe de Travail en Modélisation Géométrique (GTMG2004), 10 et 11 mars, Lyon, France, pp47-57.

[5] T. Brox, A. Bruhn, N. Papenberg and J. Weickert (2004), "High Accuracy Optical Flow Estimation Based on a Theory for Warping", Proc. ECCV2004, LNCS 3024, pp25-36.

[6] C. De Boor (1978), "A Practical Guide to Splines". Springer Verlag.

[7] G. Farin (2002), "Curves and Surfaces for CAGD: A Practical Guide". 5^{th} Edition, Morgan Kaufmann.

[8] Fleet D.J. (1992) *Measurement of Image Velocity*, Kluwer Academic Publishers, Norwell.

[9] A.F. Frangi, W.J. Niessen and M.A. Viergever (2001), "Three-Dimensional Modelling for Functional Analysis of Cardiac Images; A Review", IEEE Transaction on Medical Imaging(TMI2001), Vol. 20, No. 1, pp2-25.

[10] A. Goshtasby (1988), "Image Registration by Local Approximation Methods", Image and Vision Computing (IVC1988), Vol. 6, No. 4, pp255-261.

[11] B.K.P. Horn and B.G. Schunck (1981), "Determining Optical Flow", *Artificial Intelligence*, 17, AI1981, pp185–204.

[12] B. D. Lucas and T. Kanade (1981), "An iterative image-registration technique with an ap-

Image	$m_{u,v}/m_s$	ϕ_{2DMag}	$\phi_{2D\theta}$	$\phi_{3D\theta}$	Density
8	4/4	$23.99\% \pm 31.52\%$	$8.48° \pm 11.79°$	$6.27° \pm 5.59°$	63.98%
9	4/4	$9.99\% \pm 13.43\%$	$3.62° \pm 7.11°$	$2.52° \pm 2.06°$	61.88%
10	4/4	$9.53\% \pm 13.71\%$	$3.66° \pm 7.88°$	$2.37° \pm 2.08°$	61.84%
11	4/4	$31.85\% \pm 44.61\%$	$10.22° \pm 15.43°$	$7.68° \pm 6.22°$	63.09%
7	5/4	$26.47\% \pm 25.94\%$	$8.03° \pm 10.60°$	$7.84° \pm 5.61°$	73.16%
8	5/4	$22.95\% \pm 29.83\%$	$8.27° \pm 11.50°$	$6.02° \pm 5.42°$	62.34%
9	5/4	$10.10\% \pm 13.72\%$	$3.63° \pm 7.34°$	$2.51° \pm 2.05°$	60.34%
10	5/4	$9.61\% \pm 13.94\%$	$3.68° \pm 8.06°$	$2.37° \pm 2.09°$	60.32%
11	5/4	$31.22\% \pm 44.44\%$	$10.19° \pm 15.49°$	$7.51° \pm 6.19°$	61.54%
7	6/4	$25.52\% \pm 25.55\%$	$7.91° \pm 10.55°$	$7.46° \pm 5.32°$	70.30%
8	6/4	$22.51\% \pm 29.38\%$	$8.21° \pm 11.39°$	$5.92° \pm 5.40°$	60.58%
9	6/4	$10.18\% \pm 13.88\%$	$3.66° \pm 7.47°$	$2.52° \pm 2.05°$	58.88%
10	6/4	$9.63\% \pm 13.86\%$	$3.70° \pm 8.07°$	$2.37° \pm 2.11°$	58.93%
11	6/4	$31.04\% \pm 44.88\%$	$10.22° \pm 15.55°$	$7.45° \pm 6.19°$	59.91%
8	4/5	$22.66\% \pm 29.59\%$	$8.18° \pm 11.39°$	$6.00° \pm 5.42°$	63.95%
9	4/5	$9.41\% \pm 12.69\%$	$3.46° \pm 6.74°$	$2.38° \pm 1.99°$	61.32%
10	4/5	$9.53\% \pm 14.40\%$	$3.69° \pm 8.12°$	$2.33° \pm 2.08°$	61.27%
8	5/5	$21.78\% \pm 28.03\%$	$7.99° \pm 11.16°$	$5.78° \pm 5.26°$	62.25%
9	5/5	$9.45\% \pm 12.93\%$	$3.45° \pm 6.97°$	$2.36° \pm 2.00°$	59.92%
10	5/5	$9.53\% \pm 14.65\%$	$3.70° \pm 8.30°$	$2.31° \pm 2.08°$	59.78%
8	6/5	$21.46\% \pm 27.75\%$	$7.93° \pm 11.08°$	$5.69° \pm 5.25°$	60.44%
9	6/5	$9.51\% \pm 13.06\%$	$3.47° \pm 7.10°$	$2.35° \pm 1.98°$	58.46%
10	6/5	$9.59\% \pm 14.72\%$	$3.73° \pm 8.39°$	$2.31° \pm 2.11°$	58.48%
9	4/6	$9.04\% \pm 12.11\%$	$3.36° \pm 6.39°$	$2.30° \pm 1.97°$	60.88%
8	5/6	$20.75\% \pm 26.47\%$	$7.70° \pm 10.84°$	$5.56° \pm 5.12°$	62.07%
9	5/6	$8.99\% \pm 12.28\%$	$3.32° \pm 6.62°$	$2.26° \pm 1.96°$	59.49%
10	5/6	$9.75\% \pm 15.51\%$	$3.79° \pm 8.56°$	$2.34° \pm 2.13°$	59.39%
9	6/6	$9.00\% \pm 12.37\%$	$3.33° \pm 6.75°$	$2.24° \pm 1.94°$	58.08%
10	6/6	$9.78\% \pm 15.60\%$	$3.81° \pm 8.65°$	$2.33° \pm 2.14°$	58.10%

Table 3: Error statistics for the **Diverging Tree** sequence for images 7-11. For images 7 and 8 as well as 10 and 11 the error measures are difference measures from the correct flow field, which is known for image 9 only.

plication to stereo vision", DARPA *Image Understanding Workshop*, IUW1981, pp. 121–130 (SEE also *IJCAI1981*, pp. 674-679).

[13] J. Moore, M. Drangova, M. Wiergbicki, J. Barron and T. Peters (2003), "A High Resolution Dynamic Heart Model", Medical Image Computing and Computer-Assisted Intervention (MICCAI2003), volume 1, pp549–555.

[14] M. Otte and H.-H. Nagel (1994), "Optical Flow Estimation: Advances and Comparisons", ECCV2004, pp51-60.

[15] S. Pollard, J. Porrill and N. Thacker (1999), "TINA Programmer's Guide", Medical Biophysics and Clinical Radiology, University of Manchester, UK. (`www.niac.man.ac.uk/Tina/docs/programmers_guide/programmers_guide.html`).

[16] E.P. Simoncelli, "Design of multi-dimensional derivative filters", IEEE Int. Conf. Image Processing", Vol. 1, pp. 790-793, ICIP1994.

[17] H. Spies and J. L. Barron (2004), "Evaluating Certainties in Image Intensity Differentiation for Optical Flow", accepted to the Canadian Conference on Computer and Robot Vision (CRV2004), May.

[18] D. Suter (1994), "Motion Estimation and Vector Splines", Proc. of IEEE Computer Vision and Pattern Recognition conference (CVPR1994), pp939-948.

[19] R. Szeliski and J. Couchlan (1997), "Spline-Based Image Registration", Intl. Journal of

Image	$m_{u,v}/m_s$	ϕ_{2DMag}	$\phi_{2D\theta}$	$\phi_{3D\theta}$	Density
6	4/4	$34.16\% \pm 35.24\%$	$13.63° \pm 26.32°$	$10.65° \pm 14.58°$	58.32%
7	4/4	$20.97\% \pm 27.07\%$	$8.95° \pm 24.03°$	$7.08° \pm 13.12°$	57.43%
8	4/4	$21.45\% \pm 30.32\%$	$9.12° \pm 24.41°$	$7.13° \pm 13.37°$	57.37%
9	4/4	$22.29\% \pm 35.20\%$	$9.15° \pm 24.47°$	$7.24° \pm 13.30°$	57.65%
10	4/4	$40.53\% \pm 48.70\%$	$15.12° \pm 27.62°$	$11.83° \pm 15.21°$	58.77%
6	5/4	$32.52\% \pm 34.26\%$	$12.87° \pm 25.53°$	$9.87° \pm 13.61°$	57.43%
7	5/4	$20.16\% \pm 26.55\%$	$8.55° \pm 23.56°$	$6.63° \pm 12.47°$	56.70%
8	5/4	$20.62\% \pm 29.72\%$	$8.73° \pm 23.96°$	$6.69° \pm 12.76°$	56.63%
9	5/4	$21.46\% \pm 34.31\%$	$8.81° \pm 24.14°$	$6.82° \pm 12.76°$	56.82%
10	5/4	$39.12\% \pm 47.26\%$	$14.60° \pm 27.13°$	$11.23° \pm 14.54°$	58.00%
6	6/4	$31.50\% \pm 33.69\%$	$12.39° \pm 24.98°$	$9.35° \pm 12.90°$	56.65%
7	6/4	$19.54\% \pm 26.06\%$	$8.23° \pm 23.12°$	$6.26° \pm 11.89°$	55.91%
8	6/4	$20.01\% \pm 29.16\%$	$8.45° \pm 23.56°$	$6.35° \pm 12.26°$	55.89%
9	6/4	$20.79\% \pm 33.40\%$	$8.54° \pm 23.79°$	$6.48° \pm 12.25°$	56.13%
10	6/4	$38.17\% \pm 46.31\%$	$14.24° \pm 26.70°$	$10.77° \pm 13.88°$	57.16%
7	4/5	$20.08\% \pm 26.75\%$	$8.63° \pm 23.97°$	$6.72° \pm 12.91°$	56.88%
8	4/5	$20.45\% \pm 29.59\%$	$8.78° \pm 24.31°$	$6.75° \pm 13.11°$	56.74%
9	4/5	$21.51\% \pm 34.60\%$	$8.91° \pm 24.50°$	$6.91° \pm 13.12°$	57.06%
7	5/5	$19.20\% \pm 26.12\%$	$8.20° \pm 23.45°$	$6.24° \pm 12.20°$	56.06%
8	5/5	$19.65\% \pm 28.99\%$	$8.42° \pm 23.86°$	$6.33° \pm 12.53°$	56.00%
9	5/5	$20.64\% \pm 33.66\%$	$8.55° \pm 24.11°$	$6.49° \pm 12.59°$	56.28%
7	6/5	$18.59\% \pm 25.62\%$	$7.92° \pm 23.06°$	$5.90° \pm 11.71°$	55.28%
8	6/5	$19.10\% \pm 28.51\%$	$8.14° \pm 23.47°$	$6.01° \pm 12.06°$	55.27%
9	6/5	$19.92\% \pm 32.62\%$	$8.27° \pm 23.77°$	$6.14° \pm 12.06°$	55.47%
7	4/6	$19.65\% \pm 26.57\%$	$8.43° \pm 23.91°$	$6.52° \pm 12.72°$	56.33%
8	4/6	$19.86\% \pm 29.18\%$	$8.57° \pm 24.26°$	$6.53° \pm 12.95°$	56.24%
9	5/6	$18.74\% \pm 25.97\%$	$8.01° \pm 23.40°$	$6.05° \pm 12.06°$	55.53%
8	5/6	$19.05\% \pm 28.43\%$	$8.21° \pm 23.80°$	$6.12° \pm 12.40°$	55.50%
9	5/6	$20.28\% \pm 33.08\%$	$8.42° \pm 24.08°$	$6.33° \pm 12.45°$	55.74%
8	6/6	$18.50\% \pm 27.76\%$	$7.94° \pm 23.43°$	$5.81° \pm 11.95°$	54.78%
9	6/6	$19.56\% \pm 32.16\%$	$8.14° \pm 23.75°$	$5.97° \pm 11.91°$	54.91%

Table 4: Error statistics for the **Yosemite Fly-Through** sequence for images 6-10. For images 6 and 7 as well as 9 and 10 the error measures are difference measures from the correct flow field, which is known for image 8 only.

Computer Vision (IJCV1995), Vol. 22, No. 3, pp199-218.

[20] A. Rockwood and P. Chambers (1996), "Interactive curves and surfaces", Morgan Kaufmann.

[21] M. Unser (1999), "Splines: A Perfect Fit for Signal/Image Processing", IEEE Signal Processing magazine, pp22-38.

Image	$m_{u,v}/m_s$	ϕ_{2DMag}	$\phi_{2D\theta}$	$\phi_{3D\theta}$	Density
32	4/4	$47.72\% \pm 27.00\%$	$25.52° \pm 14.18°$	$22.94° \pm 12.21°$	62.06%
33	4/4	$42.41\% \pm 18.70\%$	$22.94° \pm 12.82°$	$20.71° \pm 11.47°$	61.17%
34	4/4	$40.60\% \pm 16.05\%$	$20.15° \pm 13.00°$	$19.24° \pm 11.28°$	61.50%
35	4/4	$36.34\% \pm 19.93\%$	$18.43° \pm 13.80°$	$17.48° \pm 12.22°$	62.02%
32	5/4	$47.35\% \pm 26.56\%$	$25.41° \pm 13.85°$	$22.74° \pm 11.95°$	60.84%
33	5/4	$42.26\% \pm 18.53\%$	$22.89° \pm 12.63°$	$20.61° \pm 11.30°$	60.14%
34	5/4	$40.44\% \pm 15.86\%$	$20.09° \pm 12.86°$	$19.14° \pm 11.14°$	60.44%
35	5/4	$35.92\% \pm 19.33\%$	$18.28° \pm 13.46°$	$17.26° \pm 11.97°$	60.82%
36	5/4	$51.07\% \pm 22.37\%$	$25.39° \pm 17.49°$	$24.89° \pm 14.30°$	70.10%
32	6/4	$47.04\% \pm 26.34\%$	$25.28° \pm 13.63°$	$22.54° \pm 11.76°$	59.80%
33	6/4	$42.04\% \pm 18.44\%$	$22.79° \pm 12.51°$	$20.47° \pm 11.18°$	59.19%
34	6/4	$40.21\% \pm 15.76\%$	$19.99° \pm 12.76°$	$19.00° \pm 11.05°$	59.50%
35	6/4	$35.55\% \pm 18.93\%$	$18.12° \pm 13.26°$	$17.07° \pm 11.82°$	59.74%
36	6/4	$50.11\% \pm 21.79\%$	$24.96° \pm 16.71°$	$24.29° \pm 13.80°$	68.16%
33	4/5	$41.89\% \pm 18.71\%$	$22.77° \pm 12.65°$	$20.47° \pm 11.35°$	60.34%
34	4/5	$39.84\% \pm 15.83\%$	$19.94° \pm 12.83°$	$18.92° \pm 11.21°$	60.66%
35	4/5	$36.78\% \pm 19.98\%$	$18.57° \pm 13.73°$	$17.65° \pm 12.19°$	61.65%
33	5/5	$41.77\% \pm 18.57\%$	$22.73° \pm 12.51°$	$20.38° \pm 11.21°$	59.34%
34	5/5	$39.71\% \pm 15.66\%$	$19.89° \pm 12.70°$	$18.83° \pm 11.09°$	59.62%
35	5/5	$36.35\% \pm 19.38\%$	$18.43° \pm 13.44°$	$17.43° \pm 11.96°$	60.42%
33	6/5	$41.57\% \pm 18.51\%$	$22.64° \pm 12.41°$	$20.24° \pm 11.10°$	58.46%
34	6/5	$39.49\% \pm 15.58\%$	$19.79° \pm 12.61°$	$18.70° \pm 10.99°$	58.74%
35	6/5	$35.97\% \pm 18.97\%$	$18.27° \pm 13.25°$	$17.22° \pm 11.80°$	59.33%
33	4/6	$41.54\% \pm 18.82\%$	$22.63° \pm 12.58°$	$20.29° \pm 11.27°$	59.66%
34	4/6	$39.24\% \pm 15.75\%$	$19.76° \pm 12.69°$	$18.66° \pm 11.16°$	59.94%
33	5/6	$41.44\% \pm 18.69\%$	$22.60° \pm 12.46°$	$20.21° \pm 11.15°$	58.73%
34	5/6	$39.14\% \pm 15.59\%$	$19.73° \pm 12.59°$	$18.58° \pm 11.05°$	59.00%
35	5/6	$36.65\% \pm 19.22\%$	$18.55° \pm 13.38°$	$17.57° \pm 11.93°$	60.08%
34	6/6	$38.96\% \pm 15.56\%$	$19.64° \pm 12.53°$	$18.47° \pm 10.98°$	58.18%
35	6/6	$36.25\% \pm 18.84\%$	$18.38° \pm 13.20°$	$17.34° \pm 11.77°$	58.97%

Table 5: Error statistics for the **Marble** sequence for images 32-36. For images 32 and 33 as well as 35 and 36 the error measures are difference measures from the correct flow field, which is known for image 34 only.

(a) (b) (c)

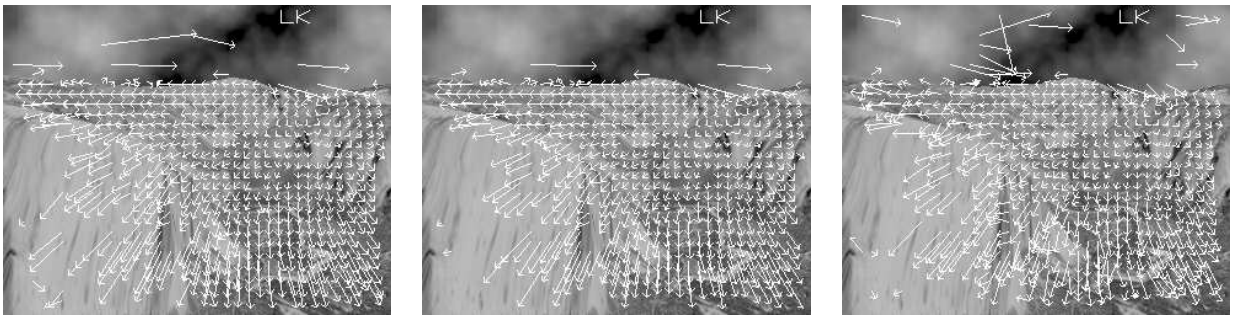
Figure 2: **Translating Tree** flows for the middle image (a) when $m_{u,v,s} = 4$, (b) when $m_{u,v,s} = 6$ and (c) when Simoncelli's filters are used.

Sequence	ϕ_{2DMag}	$\phi_{2D\theta}$	$\phi_{3D\theta}$	Density
Translating Tree	$14.12\% \pm 17.51\%$	$5.60° \pm 12.80°$	$6.16° \pm 10.99°$	55.11%
Diverging Tree	$14.05\% \pm 13.98\%$	$5.26° \pm 8.57°$	$4.10° \pm 4.69°$	60.41%
Yosemite	$32.06\% \pm 41.38\%$	$13.13° \pm 25.57°$	$11.22° \pm 15.90°$	58.32%
Marble	$42.17\% \pm 19.77\%$	$22.65° \pm 12.91°$	$20.55° \pm 11.49°$	58.94%

Table 6: Error statistics for the 4 sequences for the 9^{th} image of the **Translating** and **Diverging** Tree sequences, for the 8^{th} image of the **Yosemite Fly-Through** sequence and for the 34^{th} image of the **Marble** sequence when Simoncelli's filters were used.

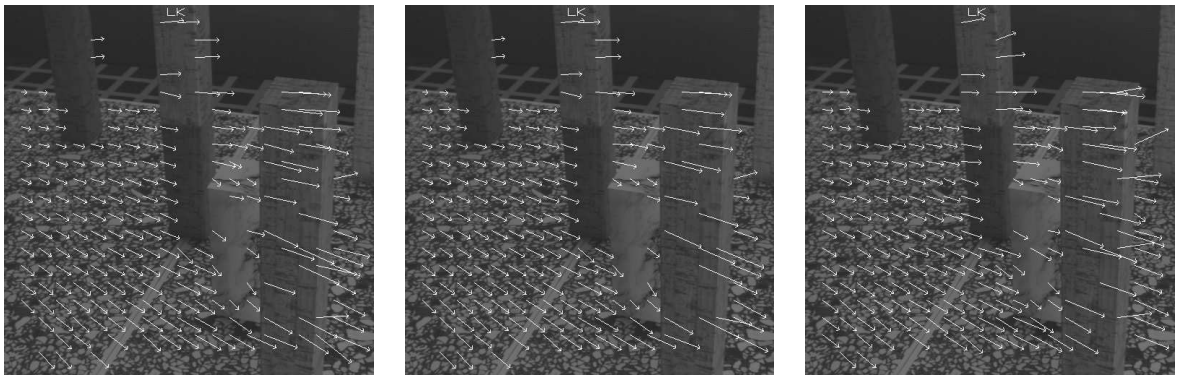
(a) (b) (c)

Figure 3: **Diverging Tree** flows for the middle image (a) when $m_{u,v,s} = 4$, (b) when $m_{u,v,s} = 6$ and (c) when Simoncelli's filters are used.



(a) (b) (c)

Figure 4: **Yosemite** flows for the middle image (a) when $m_{u,v,s} = 4$, (b) when $m_{u,v,s} = 6$ and (c) when Simoncelli's filters are used.



(a) (b) (c)

Figure 5: **Marble** flows for the middle image (a) when $m_{u,v,s} = 4$, (b) when $m_{u,v,s} = 6$ and (c) when Simoncelli's filters are used.