Image Reconstruction Error for Optical Flow

T. Lin J.L. Barron*
Department of Computer Science
University of Western Ontario
London, Ontario, Canada, N6A 5B7

Abstract

First, we examined a number of methods to perform forward and backward image reconstruction using optical flow. Given an image and its optical flow, we used these methods to generate the next image in the sequence. The RMS differences between the actual next images and their reconstructed versions for 3 synthetic image sequences, for which the correct flows were known, allowed us to determine which of our reconstruction methods performed their task well.

Second, we examined the suitability of using good image reconstruction methods as an error metric for optical flow fields computed from image sequences for which the correct flow is unknown. Again, the RMS differences between the actual next images and their reconstructed versions, which were created using the flows computed by one of 4 well known optical flow methods, were recovered for both the set of synthetic and a set of 4 real image sequences. RMS error was found to be a good indicator of optical flow error for the better reconstruction methods.

1 Introduction

2D image motion results from the projection of the 3D motion of environmental points moving relative to a sensor's image plane. Both the sensor and the environmental point are free to move independently. Optical flow (also called image velocity) is a computed approximation to this image motion based on the assumption that any changes in the spatio-temporal intensities in the sequence are due entirely to the relative motion of the sensor and the scene. The measurement of optical flow is a fundamental problem in Computer Vision and has many applications. For instance, 3D motion and structure can be inferred from 2D velocity fields [32, 21] or 2D displacement fields [20, 40] or directly from intensity derivatives [1, 28].[1] Optical flow can also be used to perform motion detection and object segmentation [5, 30, 35, 27, 38, 8, 14], to compute the focus of expansion and time-to-collision [34, 31, 39, 13], to perform motion compensated encoding [6, 26, 29] and to compute stereo disparity [3, 15, 17].

Many methods for computing optical flow have been proposed. Nine of these methods, representative of the various proposed paradigms for computing optical flow, have been examined by Barron *et al.* [4]. Their work computed optical flow using these techniques for a number of synthetic image sequences for which the correct optical flow was known and then performed a quantitative error analysis using the computed and correct flows. Flows for a number of real image sequences were also computed but since the correct flows were not available (because the correct environmental depth was not measured during the acquisition process), only a qualitative analysis was possible. We used the computed optical flow for a particular image in a sequence to compute the next image in that sequence and then compared the reconstructed image with the actual next image by computing the RMS (Root Mean Squared) difference between them [19]. Provided the image reconstruction was relatively error free, this RMS error was well correlated with actual flow error. This error metric has been used by others [29, 7, 25, 41, 24] but typically only linear interpolation was used in the reconstruction. We are interested in more accurate image interpolation that would allow good, relatively error-free, reconstruction of images with arbitrary 2D motion. Towards this goal, we examined a number of interpolation schemes for performing this task.

The interpolation methods we considered used both **forward** and **backward** projection of im-

*Please direct all enquiries to the second author, email: barron@csd.uwo.ca, phone: (519) 679-2111, ext 6896, fax: (519) 661-3515.

[1] Direct motion and structure imposes a rigidity assumption on the interpretation of spatio-temporal intensity derivatives, implicitly imposing that assumption on any computation of image velocity from the resulting motion and structure parameters.

age intensities, based on the image velocities measured at grid locations in an image sequence.[2] We used 4 optical flow methods to generate flows for 3 synthetic and 4 real image sequences and used these flows to reconstruct the next images in the sequences. Comparison of the reconstructed and actual next images was performed by computing the RMS error

$$\text{RMS error} = \sqrt{\frac{\sum_x \sum_y (I(x,y,t) - I'(x,y,t))^2}{M \times N}}, \tag{1}$$

where $I(x,y,t)$ and $I'(x,y,t)$ are the actual and reconstructed images of size $N \times M$ at time t. For the synthetic image sequences we also had the correct flows and we used Fleet and Jepson's angle error [10] to measure the flow error. Treating velocity as a spatio-temporal vector $\vec{\mathbf{v}} = (u,v,1)$ in units of (pixel,pixel,frame) we measured angle error as

$$\Phi = \cos^{-1}(\vec{\mathbf{v}}_c \cdot \vec{\mathbf{v}}_e), \tag{2}$$

where Φ is the angle between the correct spatio-temporal vector $\vec{\mathbf{v}}_c = \frac{(u_c,v_c,1)^T}{\sqrt{u_c^2+v_c^2+1}}$ and the estimated spatio-temporal vector $\vec{\mathbf{v}}_e = \frac{(u_e,v_e,1)^T}{\sqrt{u_e^2+v_e^2+1}}$ and u and v are the x and y components of an image velocity. Below, we show that angle error is well correlated with RMS error for the better reconstruction methods for the synthetic image sequences. We also demonstrate that RMS error can be a good measure of optical flow error in real image sequences with unknown correct flow.

2 Computing Optical Flow

As noted above, we considered a number of optical flow methods which were found to be accurate and had other interesting features (such as a way of computing confidence measures or certainty factors on the image velocities). We used the **local differential** method proposed by Lucas and Kanade [23, 22] with modifications proposed by [9], the **global differential** method proposed by Horn and Schunck [12], the **correlation** method of Singh [36, 37] and the **phased based** frequency method of Fleet and Jepson [10, 11]. Below, we describe the 4 methods briefly, further details are in the original citations or in Barron *et al.* [4].

- **Lucas and Kanade**: Our implementation presmoothes the image sequence by using a spatio-temporal Gaussian with a standard deviation of $\sigma = 1.5$ and performs spatio-temporal differentiation using 4-point central differences. A full velocity is computed at each image grid point from a weighted least squares fit of the normal velocities about each point to a full velocity at that point. Normal velocities are themselves computed from local spatio-temporal intensity derivatives. The smallest eigenvalue, λ_2, of the least squares matrix serve as confidence measures on the reliably of the computed velocities [9].

- **Horn and Schunck**: Presmoothing and differentiation are performed as described above for Lucas and Kanade. Instead of imposing a constant local model of velocity on the normal velocities however, Horn and Schunck impose a global smoothness constraint. That is, velocities are assumed to vary smoothly everywhere. A functional incorporating this constraint is regularized by solving a set of iterative equations. Barron *et al.* used the spatial gradient of the image intensity, ∇I, as a confidence measure [4].

- **Singh**: Singh proposes a 2-step correlation based computation. The first step uses simple correlation to compute a sum of squared differences (SSD) surface. The SSD values are converted to a probability distribution (via an exponential) and velocities are computed as weighted averages of these values. The smallest eigenvalue, λ_1, of a covariance matrix provide confidence measures. Singh's second step imposes a smoothness constraint on the velocities found in the first step. Smoothing of step 1 SSD velocities is carried out using iterative equations. Again, covariance matrix eigenvalue, λ_1, provide confidence measures. Singh also employs a coarse-to-fine velocity strategy in a Laplacian pyramid to handle fast motions and gain computational efficiency. We only use the lowest level of the pyramid in our implementation [4] as the motions in our test image sequences are relatively slow.

- **Fleet and Jepson**: Their method computes spatio-temporal phase gradients from a large number of thresholded Gabor filter responses, where each filter is tuned to a different set of spatio-temporal frequencies. These phase gradients allow component velocities[3] to be recovered. The component velocities in local neighbourhoods are fit to a 1^{st} order local least

[2] We also investigated various surface fitting schemes for reconstruction but obtained poor results.

[3] Component velocities are those normal to local phase structure while normal velocities are those normal to local gradient structure

squares velocity model. No explicit confidence measures are available as Fleet and Jepson's method is currently formulated, although they provide a thresholding mechanism based on the frequency and amplitude responses of the filters. We perform this thresholding by using $\tau = 1.25$ as in [4].

3 The Image Data

We use 3 synthetic image sequences for which the correct flows are known [4]:

- **Translating Tree** sequence: The image motion is left to right translation. This simulates a camera moving normal to its line of sight along its $x-$axis, with velocities all parallel with the image $x-$axis and having speeds between 1.73 and 2.26 pixels/frame.

- **Diverging Tree** sequence: The sensor moves along its line of sight with the focus of expansion at the centre of the image and image speeds vary from 1.29 pixels/frame on left side to 1.86 pixels/frame on the right.

- **Yosemite Fly-Through** sequence: This is a complex graphically generated image in which different parts move in different directions. The camera moves along its line of sight towards the mountain/valley (basically producing diverging flow) while the fractal-based clouds moves left to right at one pixel/frame. Speeds of up to 5 pixels/frame are exhibited in the lower left hand corner of the image.

We also use reconstruction error to investigate the optical flow error in the 4 real image sequences described in [4] for which the correct flow is unknown:

- **SRI Trees Sequence**: In this sequence the camera moves from left to right, parallel to the ground plane and perpendicular to its line of sight. Velocities are as large as 2 pixels/frame.

- **NASA Sequence**: In this sequence the camera moves along its line of sight toward the Coke can, near the centre of the image. The motion field induced by this movement is similar to that of **Diverging Tree Sequence**. Speeds are less than 1 pixel/frame.

- **Rotating Rubik Cube**: In this sequence a Rubik's cube is rotating counter-clockwise on a turntable. The velocities induced by the rotation of the cube are less than 2 pixels/frame.

- **Hamburg Taxi Sequence**: In this sequence there were four moving objects: 1) the taxi near the centre of the image turning the corner, 2) a car in the lower left, driving from left to right, 3) a van in the lower right driving from right to left and 4) a pedestrian in the upper left. Image speeds of the four objects are approximately 1.0, 3.0, 3.0 and 0.3 pixels/frame respectively.

The **Translating** and **Diverging** sequences were created by David Fleet at Queens while the **Yosemite** sequence was created by Lynn Quam at SRI. The **Nasa** and **SRI** images were obtained from the IEEE Motion Workshop Database at Sarnoff Research Centre, the **Hamburg Taxi** sequence was provided by the University of Hamburg and the **Rubik Cube** sequence was provided by Richard Szeliski at DEC.

4 Backward Reconstruction

We denote the velocity at $I(x, y, t)$, the intensity at image location (x, y) at time t, as $\mathbf{v}(x, y, t) = (u, v)$. We create a new image by using $I(x - u\delta t, y - v\delta t)$ as the intensity at location (x, y). For example, if $\mathbf{v} = (1.43, 2.31)$ at location $(50, 50)$ at time t then for $\delta t = 1$ we use the intensity at $(48.57, 47.69)$ in the image at time t as the intensity at $(50, 50)$ at time $t + 1$. The accuracy of this reconstruction depends on how precisely we can compute the subpixel intensities, i.e. the quality of the interpolation and the satisfaction of the implicit assumption that local intensity varies smoothly (this is not true, for example, at occlusion boundaries or image regions containing detailed texture). While we can compute dense flow fields from the 4 flow methods, they sometimes have thresholding capabilities, i.e. we can obtain a sparse but more accurate flow field. A question arises about how one might reconstruct parts of the image for which there is no reliable flow, because in general, good interpolation can be obtained only for dense flows. Our solution is to "fill-in" in sparse, accurate flows using linear interpolation between pairs of non-adjacent horizontal/vertical image velocities. This does introduce some errors but produces a dense and reasonably accurate flow in most cases. In general, "filled-in" flow fields are more accurate than unthresholded dense flows [18, 19]. Note that for the Lucas and Kanade and Fleet and Jepson flows computed for the Yosemite sequence, no reliable velocities can be detected for the clouds (upper 1/3 of the image) and so we do not fill in velocities here. For pixels without flow we assume a velocity of zero, meaning the pixel intensities do not change over time. In general, the filled flow fields are more accurate than the unthresholded flow fields. They are also more dense but less accurate than the thresholded flow fields (see Table 1).

We examined 3 interpolation techniques for backward reconstruction: **bilinear interpolation**, **2D polynominal interpolation** and **bicubic spline interpolation**, to compute subpixel intensity values [33].

Bilinear interpolation uses the four pixels: (i, j), $(i+1, j)$, $(i+1, j+1)$ and $(i, j+1)$, that surround (x, y), where i is the biggest integer smaller than x and j is the biggest integer smaller than y. The intensity at subpixel location (x, y) is interpolated as

$$I(x,y) = (1-t)(1-u)I(i,j) + t(1-u)I(i+1,j) + tuI(i+1,j+1) + (1-t)uI(i,j+1), \quad (3)$$

where $t = x-i$ and $u = y-j$. Bilinear interpolation is fast but less accurate than might be desired.

To obtain more accurate interpolation, we can use higher order interpolating functions. A technique called Neville's algorithm [33] finds the coefficients of an order m polynominal that interpolates a 1D set of intensity values by constructing a binary tree bottom-up via a recursive computation. The intensity value at a parent node for some subpixel location is found by linearly interpolating the intensity values at its two children nodes. The leaf nodes contain the original intensity values at adjacent pixel locations in one of the image dimensions. After the tree construction, the intensity at the tree's root is the final interpolated intensity. To perform 2D interpolation, we perform 1D interpolations in one dimension and then perform 1D interpolations on these results in the orthogonal dimension.

In some situations, interpolating polynomials are not suitable because they exhibit an oscillatory behavior as their degree increases [16]. To overcome this oscillatory behavior and still provide a smooth interpolation, we use **Cubic spline interpolation** [33]. Then **Bicubic spline interpolation** is performed as cubic spline interpolation first in one dimension and then cubic spline interpolation of those results in the orthogonal dimension.

5 Forward Reconstruction

Forward reconstruction is an alternative to backward reconstruction. Since we have the velocities at the grid locations in an image we can project the intensities of those locations onto subpixel image locations at the next time. Thus $I(x, y, t)$ is projected to $I(x+u\delta t, y+v\delta t, t+\delta t)$. We then interpolate these image intensities at the subpixel locations onto the grid (integer) locations. We consider the use of **linear patch intensity interpolation** [33] and a modification of it that we call **linear patch displacement interpolation**.

Linear patch interpolation is used to interpolate on a mesh that is not Cartesian, i.e. that has intensity values at non-integer points in 2D space rather than at the vertices of a rectangular grid. In order to compute the intensity at each pixel location in the image via linear patch interpolation, we must find the four subpixel locations which are closest to that pixel in the top-left, bottom-left, bottom-right and top-right non-overlapping quadrants. Let $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ and (x_4, y_4) be those four subpixel locations. It is not difficult to find these four subpixel locations by searching the entire subpixel image. But this is computationally expensive. We reduce the search range by limiting the magnitude of velocity components in x or y direction to be less than or equal to S_m. This search almost always finds four subpixel locations with which to perform linear patch interpolation for some location (x, y), provided (x, y) is not within S_m of any of the flow field's boundaries. In the event the search is unsuccessful, no interpolation can be performed and the intensity at (x, y) in the previous image in simply copied to (x, y) in the reconstructed image. For successful searches, interpolation is performed by using several applications of linear interpolation on these subpixel locations.

A modification to linear patch intensity interpolation is possible that may provide improved results. Because image intensity can vary greatly in local image neighbourhoods due to, for example, texture detail, using patch interpolation schemes to interpolate image intensities in such neighbourhoods can be less accurate than desirable. On the other hand, velocities (i.e. displacements), measured in such neighbourhoods should vary smoothly almost everywhere, meaning that their interpolation should be much more accurate. We use **linear patch displacement interpolation** to obtain these interpolated displacements at integer locations. If a pixel (x, y) has a velocity (u, v) at time t then at time $t+\delta t$ subpixel $(x+u\delta t, y+v\delta t)$ has the displacement $(-u\delta t, -v\delta t)$. We use linear patch interpolation to compute the pixel displacement field. Then, we perform backward reconstruction using bicubic spline interpolation to compute the intensity values for each interpolated displacement.

6 Using Weighted Averaged Flow Fields

Since accuracy is the main attribute we desire in any image reconstruction, we can smooth the flow field to suppress noisy velocities and to de-emphasis unreliable velocities in the interpolation. We employ two smoothing/averaging methods:

1. We smooth the velocity fields using a 2D Gaussian with a standard deviation of 1.5.

2. Depending on the technique, larger or smaller confidence measures can indicate better or worse velocities. We use these confidence measures to compute weighted velocity averages

$$\bar{\mathbf{v}} = \frac{\sum_{i=1}^{9} \mathbf{v}_i E_i}{\sum_{i=1}^{9} E_i}, \quad \sum_{i=1}^{9} E_i \neq 0 \qquad (4)$$

or

$$\bar{\mathbf{v}} = \frac{\sum_{i=1}^{9} \mathbf{v}_i / E_i}{\sum_{i=1}^{9} 1/E_i}, \quad \forall E_i \neq 0. \qquad (5)$$

and use these velocities in the reconstruction process. Equation (4) is used for those confidence where bigger values indicate more reliable velocities (Lucas and Kanade or Horn and Schunck) while equation (5) is used when small confidence measures indicate more reliable velocities (Singh). Zheng and Blostein [41] compute flow using the same average velocity calculation as in equation (5) in an Anandan-like flow computation [2], in their work E_i are the scaled SSD values used in the matching.

7 Results and Discussion

Below we give the results of our image reconstruction analysis. Due to space limitations we cannot show actual flow fields, their reconstructed images or the corresponding RMS error images, these are given in [19] and a thesis [18].

7.1 Angle Error for the Synthetic Images

Table 1 shows the angle error and standard deviation for the 3 synthetic image sequences for the 4 optical flow methods. Of interest is the error introduced by "filling" in the flow field performed by Lucas and Kanade and Fleet and Jepson where thresholding produces more accurate but less dense flow fields. For the translating and diverging tree sequences, the filled flow fields have about the same accuracy but are 100% dense while the filled flows for the Yosemite Fly-Through sequence are denser (about 80% density as flow for the clouds can not be filled, leading to increased RMS error for that part of the image). Although Singh's method provides thresholding capabilities, the flows, while more accurate, are too sparse to allow filling, with the exception of the translating tree sequence. For this sequence, thresholding on eigenvalue $\lambda_1 \leq 5.0$ produced $0.72° \pm 0.75°$ error with 41.4% density. Filling produced a flow field with $1.00° \pm 1.55°$ error and 99.2% density.

7.2 Reconstruction Error for the Synthetic Image Sequences

Table 2 shows the RMS reconstruction error for the 3 synthetic image sequences for the 4 optical flow methods. We show RMS error for bilinear interpolation (BI), 2D polynomial interpolation (PI), bicubic spline interpolation (BSI), forward intensity interpolation (FII), forward displacement interpolation (FDI), bicubic spline interpolation with smoothing (BSIS), forward displacement interpolation with smoothing (FDIS), bicubic spline interpolation with weighting (BSIW) and forward displacement interpolation with weighting (FDIW).

There is reconstruction error present even when the correct flow fields are used in the reconstruction although it is significantly less than the error in reconstructed images made with noisy flows. Overall, BSIS and FDIS proved to be the best interpolation schemes when using the computed flows, with BSIW and FDIW following closely behind. There is good correlation between optical flow angle error and reconstruction error for these methods. The angle error and reconstruction error generally occur at the same image locations, with the exception of Fleet and Jepson's flow fields which are too sparse to be properly filled.

7.3 Reconstruction Error for the Real Image Sequences

RMS error for the 4 real image sequences for the 4 optical flow methods are given in Table 3 for BSIS and FDIS. The error intensity images for the SRI Trees sequence show that a lot of reconstruction error is present in the upper middle part of the sequence near the tree trunks where there is a lot of occlusion. The 4 flow techniques all have problems computing flow here. For Lucas and Kanade, as the eigenvalue threshold was increased (producing more accurate but less dense flow) RMS error actually also increased, due to filling error. In general, RMS error decreased with more accurate flows. For Singh, smoothed (step 2) flow fields allowed more accurate reconstruction and hence, less RMS error. Although Fleet and Jepson was found to be better than the other 3 techniques [4], the flow fields were so sparse (as computed) that after filling, the flow fields were less accurate and hence yielded more RMS error.

Method	Translating Tree			Diverging Tree			Yosemite		
	Ave.	St. Dev.	Density	Ave.	St. Dev.	Density	Ave.	St. Dev.	Density
L&K	1.15°	1.53°	100%	2.64°	2.81°	100%	11.99°	17.94°	100%
L&K ($\lambda_2 \geq 1.0$)	0.66°	0.67°	39.55%	1.94°	2.06°	48.18%	4.11°	9.58°	33.76%
L&K (filled)	0.60°	0.57°	100%	2.11°	2.00°	100%	6.33°	11.59°	81.25%
H&S	2.14°	3.04°	100%	2.55°	3.67°	100%	11.30°	16.34°	100%
H&S ($\nabla I \geq 5.0$)	2.05°	3.48°	58.0%	2.50°	3.89°	53.5%	5.48°	10.41°	32.9%
H&S ($\nabla I \geq 5.0$, filled)	1.85°	2.90°	100%	2.82°	4.90°	100%	8.80°	11.43°	92.4%
Singh (step 1)	1.64°	2.44°	100%	17.66°	14.22°	100%	18.22°	17.01°	100%
Singh (step 2)	1.24°	1.66°	100%	10.07°	6.56°	100%	13.16°	12.07°	100%
F&J (unfilled)	0.23°	0.19°	49.68%	0.80°	0.73°	46.54%	4.95°	12.39°	30.64%
F&J (filled)	0.27°	0.21°	100%	1.50°	2.05°	100%	13.93°	22.98°	80.80%

Table 1: Angle Errors, standard deviations and densities for Lucas and Kanade, Horn and Schunck, Singh and Fleet and Jepson for the 3 synthetic images.

	Translating Tree Sequence								
Method	BI	PI	BSI	FII	FDI	BSIS	FDIS	BSIW	FDIW
Correct Flow	2.02	1.62	1.59	3.50	1.61	N/A	N/A	N/A	N/A
L&K	2.27	1.92	1.90	7.55	1.92	1.85	1.67	1.70	1.70
L&K ($\lambda_2 \geq 1.0$)	2.10	1.68	1.67	6.04	1.75	1.63	1.64	1.64	1.67
H&S	3.81	3.41	3.40	4.96	2.37	3.42	2.20	3.23	2.21
H&S ($\nabla I \geq 5.0$)	3.70	3.39	3.36	4.39	2.21	3.35	2.11	3.25	2.16
Singh (step 1)	2.88	2.80	2.82	6.36	3.32	2.60	2.66	2.72	3.20
Singh (step 1, $\lambda_1 \leq 5.0$)	3.09	2.66	2.64	5.96	2.10	2.56	2.02	2.60	2.00
Singh (step 2)	2.67	2.60	2.58	4.45	2.40	2.34	2.38	2.47	2.46
F&J	2.08	1.66	1.64	3.46	1.64	1.62	1.60	N/A	N/A
	Diverging Tree Sequence								
Method	BI	PI	BSI	FII	FDI	BSIS	FDIS	BSIW	FDIW
Correct Flow	4.54	3.48	3.37	4.57	3.36	N/A	N/A	N/A	N/A
L&K	4.85	3.73	3.60	5.44	3.65	3.55	3.56	3.61	N/A
L&K ($\lambda_2 \geq 1.0$)	4.87	3.72	3.58	5.47	3.62	3.50	3.59	3.52	3.60
H&S	4.97	4.02	3.91	5.27	3.74	3.86	3.71	3.89	3.72
H&S ($\nabla I \geq 5.0$)	4.94	3.95	3.89	5.35	3.76	3.85	3.71	3.84	3.70
Singh (step 1)	8.88	8.09	8.10	11.86	10.15	6.93	7.11	7.62	7.98
Singh (step 2)	6.01	5.76	5.63	6.63	5.69	5.53	5.57	5.60	5.67
F&J	5.14	3.91	3.76	5.31	3.77	3.74	3.76	N/A	N/A
	Yosemite Fly-Through Sequence								
Method	BI	PI	BSI	FII	FDI	BSIS	FDIS	BSIW	FDIW
Correct Flow	6.31	5.92	5.82	6.81	5.79	N/A	N/A	N/A	N/A
L&K	6.67	6.18	6.09	8.57	6.07	5.97	5.88	6.01	5.90
L&K ($\lambda_2 \geq 1.0$)	6.77	6.44	6.42	7.56	6.29	6.27	6.19	6.29	6.19
H&S	6.90	6.81	6.71	8.14	6.62	6.58	6.38	6.59	6.39
H&S ($\nabla I \geq 5.0$)	7.01	6.60	6.51	7.18	6.38	6.41	6.17	6.44	6.18
Singh (step 1)	8.51	7.31	7.32	10.49	7.83	6.76	6.70	7.18	7.05
Singh (step 2)	7.02	6.41	6.33	7.25	6.38	6.30	6.34	6.32	6.36
F&J	8.28	8.22	8.12	8.32	7.15	7.93	7.09	N/A	N/A

Table 2: RMS errors for various interpolation schemes for the 3 synthetic image sequences for Lucas and Kanade, Horn and Schunck, Singh and Fleet and Jepson.

Method	SRI Trees		Nasa		Rubik Cube		Taxi	
	BSIS	FDIS	BSIS	FDIS	BSIS	FDIS	BSIS	FDIS
L&K	7.21	7.19	1.91	1.98	1.22	1.28	3.79	3.90
L&K ($\lambda_2 = 1.0$)	7.20	7.20	2.01	2.05	3.05	2.64	4.17	4.24
H&S	7.49	7.43	2.03	2.01	2.96	1.94	3.82	3.93
Singh (step 1)	8.16	8.07	3.88	3.91	3.23	3.50	4.11	4.27
Singh (step 2)	7.75	7.63	3.58	3.60	2.82	2.93	4.04	4.18
F&J	7.14	7.12	2.77	2.75	3.04	2.43	4.09	4.13

Table 3: RMS error for the Real Image sequences for various interpolation schemes for flows computed by Lucas and Kanade, Horn and Schunck, Singh and Fleet and Jepson.

8 Conclusions

We have shown that image reconstruction is a useful optical flow error metric. For the 3 synthetic image sequences we have shown that angle error is well correlated to RMS reconstruction error for a number of the better interpolation techniques. The error images for the angle error and the RMS error are well correlated. This demonstrates that RMS error is a good indicator of angle error. As well, we saw that using optical flow confidence measures during reconstruction was beneficial to the interpolation calculations. For the 4 real image sequences we saw that the error intensity images indicate large error at object boundaries/occlusion boundaries where image velocity computation is problematic.

Reconstruction is difficult at occlusion boundaries. For a surface A, whose motion covers another surface B (A occludes B), then forward reconstruction seems best. On the other hand, if surface A is being uncovered by the motion of surface B (B disoccludes A), then backward reconstruction is best. What is needed is a hybrid reconstruction method that detects the location and type of occlusion and uses either forward or backward reconstruction depending on which surface is being occluded or disoccluded.

References

[1] J. Aloimonous and C.M. Brown. Direct processing of curvilinear sensor motion from a sequence of prespective images. In *2nd Workshop on Visual Representation: Representation and Control*, pages 72–77, 1984.

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

[3] S. T. Barnard and W. B. Thompson. Disparity analysis of images. *IEEE PAMI*, 2(4):333–340, 1980.

[4] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *IJCV*, 12(1):43–77, 1994.

[5] M. J. Black and P. Anandan. A model for the detection of motion over time. In *ICCV*, pages 33–37. IEEE, 1990.

[6] B. Capentieri and J.A. Storer. A split-merge parallel block-matching algorithm for video displacement estimation. In *Data Compression Conference*, pages 239–248, 1992.

[7] E. Dubois. The sampling and reconstruction of time-varying imagery with application in video systems. *Proc. of IEEE*, 73(4):502–522, 1985.

[8] J. H. Duncan and T. Chou. On the detection of motion and the computation of optical flow. *IEEE PAMI*, 14(3):346–352, 1992.

[9] E.H. Adelson E.P. Simoncelli and D.J. Heeger. Probability distributions of optical flow. In *IEEE Proc. of CVPR*, pages 310–315, 1991.

[10] D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *IJCV*, 5(1):77–104, 1990.

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

[12] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–204, 1981.

[13] R. C. Jain. Direct computation of the focus of expansion. *IEEE PAMI*, 5(1):58–63, 1983.

[14] R. C. Jain. Segmentation of frame sequences obtained by a moving observer. *IEEE PAMI*, 6(5):624–629, 1984.

[15] M. R. M. Jenkin, A. D. Jepson, and J. K. Tsotsos. Techniques for disparity measurement. *CVGIP*, 53(1):14–30, 1991.

[16] L.W. Johnson and R.D. Riess. *Numerical Analysis.* Addison-Wesley, 1982.

[17] K. Langley, T. J. Atherton, R. G. Wilson, and M. H. E. Lacombe. Vertical and horizontal disparities from phase. *Image and Vision Computing*, 9(4):296–302, 1991.

[18] T. Lin. Image reconstruction error for optical flow. Master's thesis, University of Western Ontario, Dept. of Computer Science, December 1993.

[19] T. Lin and J.L. Barron. Image reconstruction error for optical flow. Technical Report TR-412, Univ. of Western Ontario, Dept. of Computer Science, 1994.

[20] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 223:133–135, 1981.

[21] H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proc. R. Soc. Lond.*, B 208:385–397, 1980.

[22] B. D. Lucas. *Generalized image matching by the method of differences.* PhD thesis, Carnegie-Mellon Univ., 1984.

[23] B. D. Lucas and T. Kanade. An iterative image-registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130. DARPA, 1981.

[24] M.R. Luettgen, W.C. Karl, and A.S. Willsky. Efficient multiscale regularization with applications to the computation of optical flow. *IEEE Trans. on Image Processing*, 3(1):41–64, January 1994.

[25] H.G. Mausmann, P. Pirsch, and H.J. Grallert. Advances in picture coding. *Proc of IEEE*, 73(4):523–548, 1985.

[26] F.W. Mounts. A video encoding system using conditional picture-element replenishment. *Bell System Technical Journal*, pages 2545–2554, 1969.

[27] D. W. Murray and B. F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE PAMI*, 9(2):116–180 OR IS IT 220–228, 1987.

[28] S. Negahdaripour and B. K. P. Horn. Direct passive navigation. *IEEE PAMI*, 9(1):168–176, 1987.

[29] A.N. Netravali and J.D. Robbins. Motion compensated television coding: Part 1. *Bell System Technical Journal*, pages 631–670, 1979.

[30] M. Ogata and T. Sato. Motion-detection model with two stages: Spatiotemporal filtering and feature matching. *J. Opt. Soc. Am.*, A 9(3):377–387, 1992.

[31] I. Overington. Gradient-based flow segment and location of the focus of expansion. *Alvey Vision Conference*, pages 860–870, 1987.

[32] K. Prazdny. Motion and structure from optical flow. In *IJCAI*, pages 702–704, 1979.

[33] W.H. Press, B.P. Flannery, S.A. Teudilsky, and W.T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.

[34] D. Regan and K.I. Beverley. How do we avoid confounding the direction we are looking and the direction we are moving. *Science*, 215:194–196, 1982.

[35] W. Reichardt, R. W. Schlogl, and M. Egelhoaf. Movement detectors of the correlation type provide sufficient information for local computation of 2d velocity fields. *Naturwissenschaften*, 75:313–315, 1988.

[36] A. Singh. An estimation-theoretic framework for image flow computation. In *ICCV*, pages 168–177, Osaka, Japan, 1990. IEEE.

[37] A. Singh. *Optic flow computation: A unified perspective.* IEEE Computer Society Press, 1992.

[38] L. A. Spacek. Edge detection and motion detection. *Image and Vision Computing*, 4(1):43–56, 1986.

[39] M. Subbarao. Bounds on time-to-collision and rotational component from first-order derivatives of image flow. *CVGIP*, 50:329–341, 1990.

[40] R. Y. Tsai, T. S. Huang, and W. Zhu. Estimating three-dimensional motion parameters of a rigid planar patch 2: Singular value decomposition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 30(4):525–534, 1982.

[41] H. Zheng and S.D. Blostein. An error-weighted regularization algorithm for image motion-field estimation. *IEEE Trans. on Image Processing*, 2(2):246–252, 1993.