Sequence	Algorithm	Avg. Error	Std. Dev.	Density
Translating Tree	Horn & Schunck	33.4	16.46	100
	Heeger	4.79	2.39	13.8
	Anandan	4.54	2.98	100
	Lucas & Kanade (λ_2 >1.0)	1.75	1.43	40.8
	Fleet & Jepson	0.36	0.41	76.0
	Weber & Malik	0.49	0.35	96.8
Diverging Tree	Horn & Schunck	9.85	8.86	100
	Heeger	4.95	3.09	73.8
	Anandan	8.23	6.17	100
	Lucas & Kanade (λ_2 >1.0)	3.05	2.53	49.4
	Fleet & Jepson	1.24	0.72	64.3
	Weber & Malik	3.18	2.50	88.6
Yosemite	Horn & Schunck	22.58	19.73	100
	Heeger	11.74	19.00	44.8
	Anandan	15.54	13.46	100
	Lucas & Kanade (λ_2 >1.0)	5.20	9.45	35.1
	Fleet & Jepson	4.29	11.24	34.1
	Weber & Malik	3.43	5.35	45.2

Figure 7: Comparison of synthetic sequences results with those reported by Barron et. al. [BFBB92]. The Weber & Malik algorithm uses only 10 frames and 30 linear filters. The Fleet & Jepson algorithm used 46 3-d convolutions and 21 frames (15 frames for Yosemite).

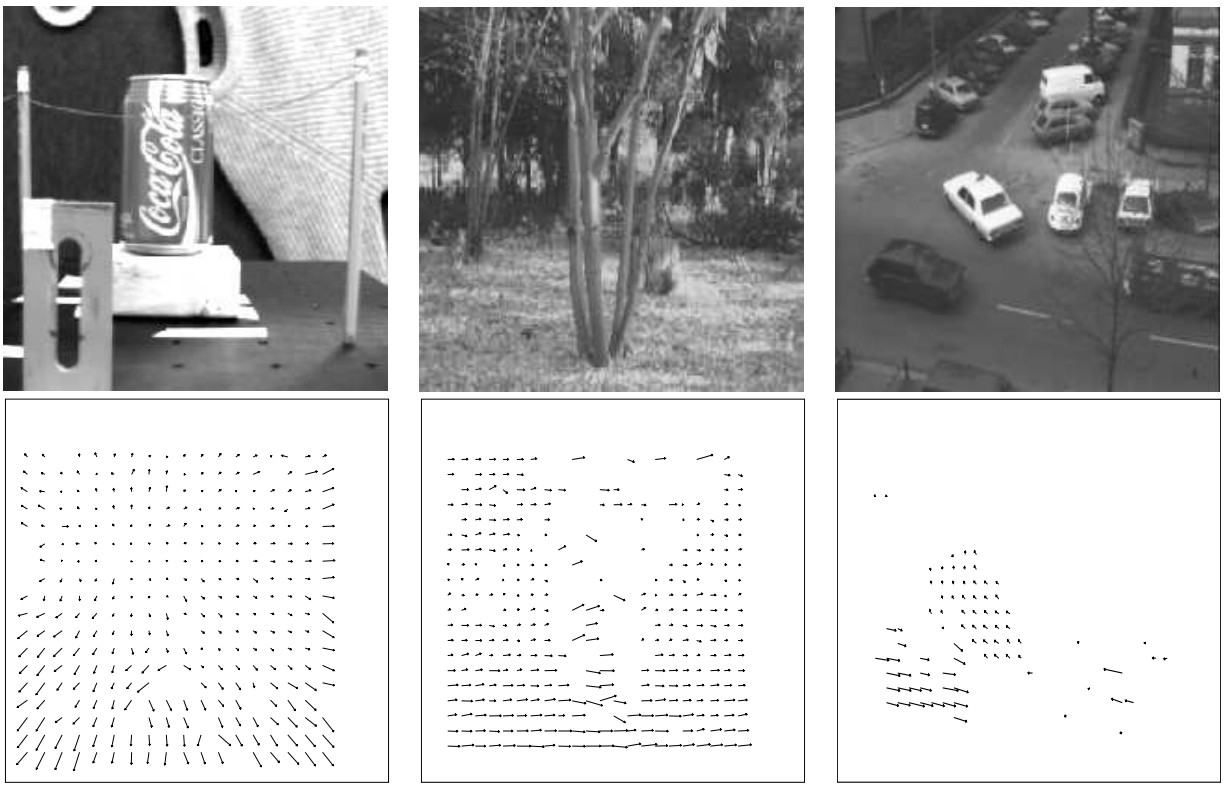

Figure 8: Three frames from real image sequences and the flow recovered.

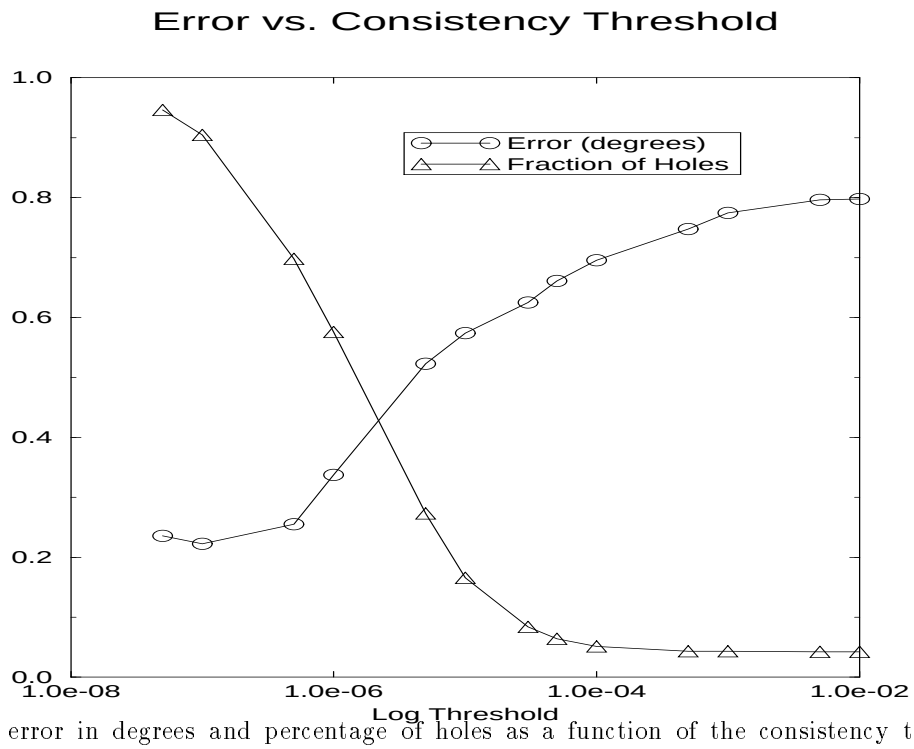

Figure 6: Average error in degrees and percentage of holes as a function of the consistency threshold, C_t, the maximum of the ratio σ_3/σ_2 allowed. This ratio relates how much the measurements must be changed in order to fit the velocity model.

Fleet & Jepson algorithm used 46 3-d convolutions and 21 frames making it computationally more expensive. In addition, our algorithm consistently produced a higher density of vectors. The threshold experiments show that slight improvements can be made by decreasing the error threshold (fixed at 10^{-2} for synthetic sequences) at a cost of fewer estimates. Ultimately, the performance must be based on how well the flow field can be used for calculating quantities such as motion and shape parameters.

5.2 Real Sequences

The algorithm was tested on a group of real video images obtained from J.L. Barron who received them from the database at Sarnoff Research Centre. We used 10 frames and 25 filters for each. Selected frames of the three sequences and the flow produced are shown in Figure 8. The first sequence is of the camera translating towards the soda can. In the second, the observer translates perpendicular to the line of sight. The tree in the foreground translates more due to perspective. The third sequence is of three independently moving cars. The car on the lower right is obscured by some trees.

5.3 Subsampled Filters

The largest scales consists of filters with Gaussian responses many pixels in width. The response of such filters does not change significantly within a range of a few pixels. The full version of the algorithm computes the response of every filter at each pixel. Since the large scale filters do not vary much a more efficient implementation would use a pyramid scheme in which filters of larger scales are calculated at a subset of the full pixel lattice. We created a modified version of the algorithm in which the response of a filter with Gaussian response of size σ is subsampled every n pixels, where $n = \lfloor \sigma \rfloor$. This reduced by about half the number of convolutions for each scale from the previous scale. The results of the subsampled version of the algorithm for the synthetic images are tabulated in Table 9. The flow field became blocky in this scheme. The component values remained almost constant within small sub-blocks of pixels, especially for large displacements, where only large scale filters could make valid estimates. Here the estimate must be the same for the entire sub-block of the smallest filter used. Qualitatively the flow was still very good, but consisted of blocks of pixels with the same flow estimate.

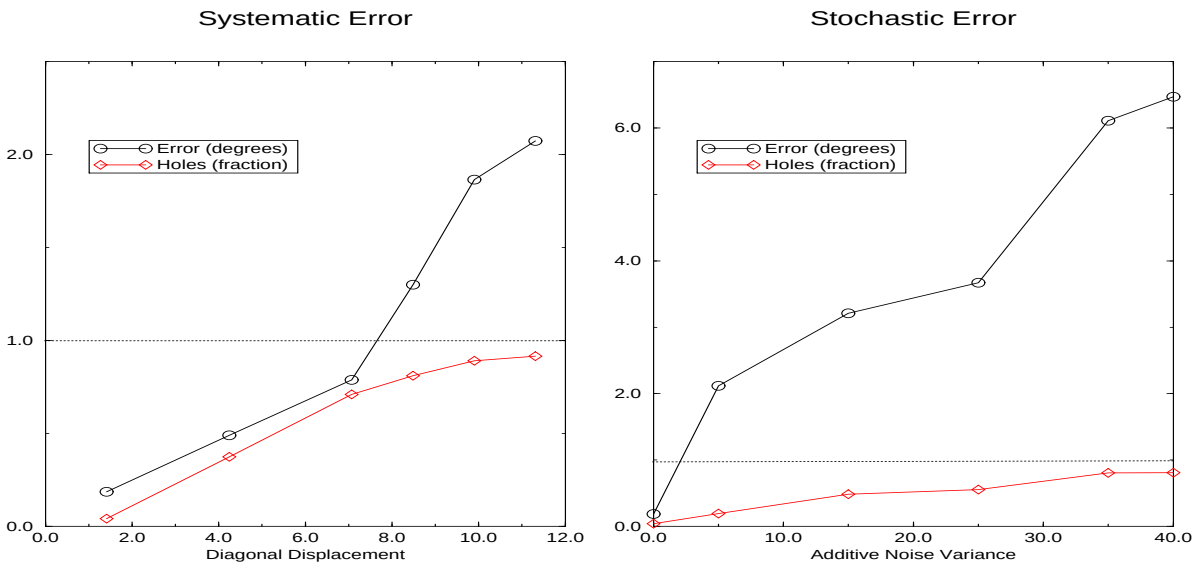

Figure 5: Systematic error due to increasing displacements with no noise, and stochastic error due to additive noise with fixed dispacement. In both cases the stimulus was a Gaussian white noise pattern with variance 1000 units translated diagonally. The consistency threshold was set at 1×10^{-2}. This value keeps the errors to only a few degrees for the full range of allowed displacements.

as a function of diagonal displacement. As the displacement increases the average error increases, but remains less than two degrees. The number of estimates decreases up to the maximum displacement allowed, $1.8^4 \simeq 10.5$ pixels. Thus, up to the maximum allowed displacement, we can keep errors to less that a few degrees. A higher density of estimates can be obtained by lowering the consistency threshold at a cost of increased errors.

Stochastic Error. In order to examine the results of stochastic error we translated the uncorrelated random dot pattern diagonally 1 pixel and added uncorrelated noise to each frame. Figure 5 shows the results as the standard deviation of the added Gaussian noise is increased. Again, the errors remain less than a few degrees as the density decreases.

Consistency Threshold. A random dot pattern of unit variance was translated diagonally 1 pixel and white noise of variance 5 units was added to each frame. Figure 6 displays the average error and percentage of holes as a function of C_t, the equation consistency threshold. The number of estimates decreases and the accuracy of the remaining estimates increases until only a few estimates remain. This demonstrates that the singular values ratio is a good measure of estimate reliability. Such a statistic is often useful in algorithms which use optical flow as input, such as for determining ego-motion or shape from motion.

Comparison Sequences. A recent report by Barron et. al. [BFB92] and the corresponding paper in CVPR '92 [BFBB92] examined the performance of a number of different optical flow techniques on a series of synthetic and real images. They found that the phase-based approach of Fleet & Jepson [FJ90] was the most accurate. They also felt that Heeger's energy approach [Hee87] and other's based on SSD minimization were computationally prohibitive. We examined the performance of our routine on these same images.

Ten frames of the *Yosemite Sequence* were used as input to the program. The true optical flow is known because this is a syntheticly generated sequence. The sequence is of a platform flying over Yosemite valley. Clouds in the image deform as they move. The optical flow ranged in magnitude from zero at the focus of expansion to over 5 pixels per frame. The *Translating Tree* sequence was consists of a tilted plane with a texture mapped onto it. The motion is perpendicular to the optical axis, but since the plane is tilted the flow ranged in magnitude from 1.8 to 2.3 pixels per frame. The *Diverging Tree* consisted of the same tilted plane and texture, but the motion is along the optical axis. Velocities ranged from 1.4 pixels per frame on one side to 2.0 on the other. Table 7 lists the average and standard deviation of the angular error. The data for the other algorithms was copied from the CVPR paper by Barron et al[BFBB92]. The performance was comparable, performing better for the Yosemite sequence and worse on the translating planes. However, our algorithm uses only 10 frames and 30 linear filters whereas the