February 15, 1994 CS-ETR-94-01

Spectrum selective techniques for optical flow estimation:

Minas E. Spetsakis

Dept. of Computer Science
York University
4700 Keele Street
North York, ONTARIO
CANADA, M3J 1P3

ABSTRACT

The computation of the optical flow using least squares needs a weighting scheme to ensure that every constraint is taken into account with weight proportional to its reliability and the closely correlated constraints are not considered many times. This paper presents a method that introduces a simple weighting scheme using a convolution to the least squares minimization. We show how this technique can be applied in conjunction with several other techniques in the literature and we apply the technique for the case of non uniform motion when filters with large support are used (as is the case of hierarchical flow estimation). The method is tested on real and synthetic images.

The support of the NSERC (App. No. OGP0046645) is gratefully acknowledged.

1. Introduction

The estimation of the flow field of a moving image is an intermediate step for the structure from motion problem. Although it has been shown that in many cases the optic flow is an overkill and we can do scene recovery without it [14, 13], any advances in the area of flow estimation will have direct consequences on most of the field of structure from motion. This is the reason that there so much work has been done and so many approaches proposed [7, 4, 9, 10, 11, 15] for the general solution of the problem. There are also several approaches to the problem under adverse conditions like transparency [8, 3].

A wide class of approaches to the flow problem follows the following pattern. First a set of constraints is established based on first principles or intuition. Then these constraints are expressed as a least squares partial differential equation and the equation is then solved. While the problem is underconstrained in its basic form, with additional constraints derived from the application domain of the algorithm becomes overconstrained. Quite often the equations are weighted according to the reliability of each constraint for each pixel. Such a weighting silently assumes that the residual error of the equations is uncorrelated among neighboring pixels.

In this paper a weighting scheme is proposed to take care of the interdependence of the constraints among neighboring pixels. The technique works by selectively attenuating the noisier parts of the spectrum of the residual. We show how this method can be integrated with other methods from the literature and we design filters that take care of the interdependence that arises when the flow is non uniform and filtering with wide support is being used (as when we filter to get a pyramid of resolutions).

The organization of the rest of the paper is as follows. Sections 2 and 3 give the motivation for this approach by describing two cases where the residual is not independent among pixels. Section 4 describes the technique in its basic form. Section 5 shows how this method can be combined with several others. Section 6 deals with the selection of filters that minimize the effects of spectral warping. Section 7 contains experimental results with real and synthetic images. Section 8 concludes the paper.

2. Spectrum Warping

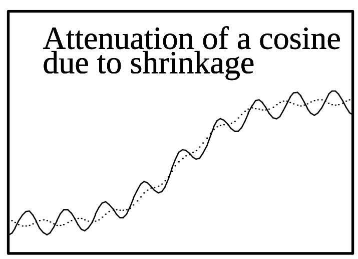
The flow of a moving image is hardly ever uniform. There are always regions that rotate, dilate, sheer, move uniformly or just don't move. There are also discontinuities in the flow field where the flow cannot be meaningfully defined in which case the quality of an algorithm can be judged only by how reliably it detects discontinuities and how close to the discontinuity it can estimate the flow.

Uniform motion and no motion are fairly easy to handle. The problems start with non uniform motion. So let's examine it more closely.

The non uniform motion in a small region can be considered as a superposition of a uniform flow, a rotation, a dilation (may be different dilation along the two axes) and a sheer. The Fourier transform of such an image is transformed itself by this flow. Due to the uniform motion it undergoes a phase shift. Due to rotation, rotates. And due to dilation, contracts (negative dilation).

The problems start when we filter the image (the camera itself filters the image with a lowpass filter otherwise would see infinite detail, or have infinite aliasing). Assume we filter the image with a sharp lowpass filter (Fig. 2.1). Further assume that the image contains a cosine which is just below the cutoff frequency. Then the cosine is still visible after the filtering. But if the image contracts then in a subsequent frame the cosine will move beyond the cutoff frequency and it will become invisible. Any flow algorithm will have real problems with this. The residual error in the optical flow equation will be substantial.

But this is not the end of the world. This residual is mainly composed of frequencies around the cutoff. The assumption behind the idea of minimizing the sum of the squares of the residual is that the spectrum of the residual is flat i.e. white noise. When a particular area of the spectrum is more “noisy” than the others then the neighboring points are statistically



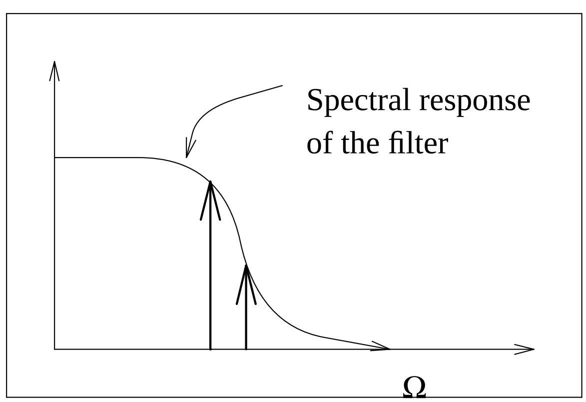


Figure 2.1 The figure shows the image of a cosine. The image shrunk as it moved, and the cosine went beyond the cutoff point of the filter and was attenuated.

correlated.

A flow algorithm that uses straight least squares will try to reduce this residual uniformly. It would be better though, to give less weight to the components of the spectrum that are more noisy. To achieve this we convolve the residual with a filter that cuts the suspicious frequencies around the cutoff point of the filter.

3. Changing lighting conditions

A common problem with the gradient based flow algorithms is the changing lighting conditions. They can affect the algorithm seriously. One way out is to use the logarithm of the intensity, in which case the lighting conditions become an additive component instead of multiplicative. In most cases the lighting conditions vary slowly in the scene (specularities are not among them), so we can use a filter to cut off all the very low frequencies. But we run into a problem similar to the above. The frequency components around the cutoff might cross the cutoff as the image changes.

The solution is to look at the residual of the optic flow equation. If the residual at a certain point has a non zero value then if this is due to the lighting conditions then the point next to it will exhibit a very similar residual. In other words the residuals at neighboring points are again correlated. The solution is now to filter the residual to cut the very low frequencies.

4. Spectrum Selective Flow Estimation

As in most flow estimation algorithms we try to minimize a measure of the residual of the optic flow equation:

$$I_{err} = I_x u + I_y v + I_t \tag{4.1}$$

This residual I_{err} is a random image (or two dimensional random process) every pixel of which is not statistically independent of the rest. If we assume that it is a stationary process though, then there is a template W such that $I_{serr} = W \otimes I_{err}$ is white noise. In other words, every pixel of the process is a random variable independent of the rest and all of them identically distributed. The $\otimes$ is the convolution operator. We try to find a u and a v that minimize the Sum of the Squared Errors (SSE).

$$SSE = \sum_{i,j} I_{serr}^2[i, j] = \sum_{i,j} \left[(I_x u + I_y v + I_t) \otimes W \right]_{i,j}^2 = \sum_{i,j} \left[(I_x u) \otimes W + (I_y v) \otimes W + I_t \otimes W \right]_{i,j}^2$$

SSE is a quadratic function in the components of the flow field u and v. To get the linear equations we take the derivatives with respect to the unknowns.

$$\rho_u[kl] = \frac{\partial SSE}{\partial u[k,l]}, \quad \rho_v[kl] = \frac{\partial SSE}{\partial v[k,l]}$$

First we compute:

$$\frac{\partial \left[(I_x u) \otimes W \right]_{i,j}}{\partial u[k,l]} = \frac{\partial \sum_{m,n} I_x[i-m, j-n] u[i-m, j-n] W[m,n]}{\partial u[k,l]} =$$

$$\sum_{m,n} I_x[i-m, j-n] \delta[i-m-k, j-n-l] W[m,n] = I_x[k,l] W[i-k, j-l]$$

where $\frac{\partial u[r,s]}{\partial u[k,l]} = \delta[r-k, s-l]$ because u and v are the independent variables of the system and $\delta[.,.]$ which is 1 at $[0,0]$ and 0 everywhere else, is Kronecker's delta. The summation was eliminated because δ is non-zero only for $m = i-k$ and $n = j-l$. In a similar way we obtain the derivative of the other term. So:

$$\rho_u = \frac{\partial SSE}{\partial u[k,l]} = \frac{\partial}{\partial u[k,l]} \sum_{i,j} \Big[(I_x u) \otimes W + (I_y v) \otimes W + I_t \otimes W \Big]^2_{i,j} =$$

$$2 \sum_{i,j} I_x[k,l] W[i-k, j-l] I_{serr}[i,j] = 2 I_x[k,l][W \ast I_{serr}]_{kl} = 2 I_x[k,l] \Big[W \ast W \otimes I_{err} \Big]_{kl} =$$

$$2 I_x[k,l][R_w \otimes I_{err}]_{kl}$$

where R_w is the autocorrelation of W and $\ast$ is the correlation operator. R_w is the spectrum selection filter.

Thus

$$\begin{aligned} \rho_u &= I_x \;\; R_w \otimes I_{err} \\ \rho_v &= I_y \;\; R_w \otimes I_{err} \end{aligned} \tag{4.2}$$

where the factor 2 was dropped because it is a common multiplier to everything. These are the two linear equations that we have to solve to find the flow. Simple inspection reveals that they are not independent equations. In the next section we show how to get more constraints considering various scales and stabilization factors like smoothness [7]. The new constraints will be added linearly on these, so the above equations will serve as the core of the system of equations.

Whether we use Jacobi iteration or Conjugate Gradient we need a preconditioner. The most convenient choice would be the diagonal or the 2×2 blocks on the diagonal. If we represent the matrix of (4.2) as an explicit matrix then getting these blocks would be straightforward. But we prefer to keep the equations in an implicit form so that computations are faster and the storage minimal. To get the diagonal elements then we use differentiation:

$$B_1 = \frac{\partial \rho_u[k,l]}{\partial u[k,l]} = \frac{\partial}{\partial u[k,l]}\,(I_x[k,l](R_w \otimes (I_x u))) = I_x^2[k,l]R_w[0,0]$$

$$B_2 = \frac{\partial \rho_u[k,l]}{\partial v[k,l]} = \frac{\partial}{\partial v[k,l]}\left(I_x[k,l](R_w \otimes (I_y u))\right) = I_x[k,l]I_y[k,l]R_w[0,0]$$

$$B_3 = \frac{\partial \rho_v[k,l]}{\partial u[k,l]} = B_2$$

$$B_4 = \frac{\partial \rho_v[k,l]}{\partial v[k,l]} = \frac{\partial}{\partial v[k,l]}\left(I_y[k,l](R_w \otimes (I_y v))\right) = I_y^2[k,l]R_w[0,0]$$

5. Combining with other techniques

One advantage of the spectrum selective technique is that it is almost orthogonal to several other techniques used in flow estimation. With only a small penalty on the CPU time we can combine a several of them, or at least all the ones that do not work opposite to each other. In this section we present versions of the spectrum selective technique combined with hierarchical methods [1], window based methods [9] etc. In the experiments section we show how all these can be combined together and discuss the results.

5.1. Second order iteration

The optic flow equation is a first order approximation of the optic flow constraint, which in most cases is not enough. There are two alternatives: one is to include higher order terms, which will clearly complicate the already difficult equations and the other is to use a guess for the flow (for the first iteration the guess is zero) to get a better estimate for the time derivative. This second approach is equivalent to applying an image warp based on the current estimate of the flow, but we prefer to avoid the expensive and sometimes unstable operation of warping.

Starting from the constancy of the intensity constraint:

$$I(x, y, t) = I(x + \Delta x + \Delta\Delta x, y + \Delta y + \Delta\Delta y, t + \Delta t)$$

where Δx, Δy are the guess for the displacement and $\Delta\Delta x$, $\Delta\Delta y$ are unknowns. Then

$$\begin{aligned} I(x, y, t) = {} & I(x, y, t) + I_x(x, y, t)\Delta x + I_x(x, y, t)\Delta\Delta x + \\ & I_y(x, y, t)\Delta y + I_y(x, y, t)\Delta\Delta y + I_t(x, y, t)\Delta t + O_2 \\ I(x, y, t) = {} & I(x + \Delta x, y + \Delta y, t + \Delta t) + I_x(x, y, t)\Delta\Delta x + I_y(x, y, t)\Delta\Delta y + O_2 \end{aligned}$$

by rearranging and ignoring the higher order terms we get:

$$I_x(x, y, t)\Delta\Delta x + I_y(x, y, t)\Delta\Delta y + (I(x + \Delta x, y + \Delta y, t + \Delta t) - I(x, y, t)) = 0$$

To this we add:

$$I_x(x, y, t)\Delta x + I_y(x, y, t)\Delta y - I_x(x, y, t)\Delta x - I_y(x, y, t)\Delta y = 0$$

and we get finally:

$$I_x u + I_y v = -I_t^b$$

where $u = \dfrac{\Delta\Delta x + \Delta x}{\Delta t}$ and $v = \dfrac{\Delta\Delta y + \Delta y}{\Delta t}$ and the better estimate of the time derivative I_t^b is:

$$\frac{(I(x + \Delta x, y + \Delta y, t + \Delta t) - I(x, y, t)) - I_x(x, y, t)\Delta x - I_y(x, y, t)\Delta y}{\Delta t}$$

If we restrict ourselves to guesses that have integer values then the cost of computation is small.

5.2. Hierarchical and window based approach

Although the window based approach dates back to early 80's [9], it outperforms many modern techniques in efficiency and stability [2]. It is definitely worth combining this method with the spectrum selective techniques.

The hierarchical approaches on the other hand are of two kinds: the first kind uses the result of the coarse levels as a guess for the finer levels [1] and the other fuses the equations from all the levels in one set of equations with different weights which are modified as the iteration progresses. This second approach is more general than the first, and we are going to use this as our model. Another reason, though that we choose this, is that it leads to more elegant and efficient solutions when combined with spectrum selective techniques and that it allows for the design of better spectrum selection filters.

For the rest of the section we show that the window based methods are in many ways similar to hierarchical methods of the second kind.

If we assume that the flow is, more or less, constant within a window $w^{\dagger}$ then

$$I_{err}[m,n] = I_x[m,n]u[i,j] + I_y[m,n]v[i,j] + I_t[m,n]$$

should be very small everywhere around $[i, j]$. So

$$err[i,j] = \sum_{m,n} w[i-m, j-n] I_{err}^2 =$$

$$\sum_{m,n} w[i-m, j-n](I_x[m,n]u[i,j] + I_y[m,n]v[i,j] + I_t[m,n])^2$$

and then we minimize the sum of the *err*'s as done in [9, 5, 6]. We find a u and a v that minimize the weighted sum of squares of the optic flow equations in a small neighborhood.

Alternatively, we can minimize the sum of squares of linear combinations of the optic flow equations in this neighborhood.

---

† Typically a window is defined as a box or a Gaussian function. In this case the window is the support of the function.

$$err[i,j] = \sum_q \left\{ \sum_{m,n} I_{err}[m,n] g_q[i-m, j-n] \right\}^2$$

$$\sum_q \left\{ \left[\sum_{m,n} I_x[m,n] g_q[i-m, j-n] \right] u[i,j] + \left[\sum_{m,n} I_y[m,n] g_q[i-m, j-n] \right] v[i,j] + \left[\sum_{m,n} I_t[m,n] g_q[i-m, j-n] \right] \right\}^2$$

where g_q are the weights of the various linear combinations. In the case of ordinary linear systems, nothing changes if we replace the equations with their linear combinations (provided that these linear combinations do not form a singular transformation). But in the case of least squares, doing this we disturb the relative weights of the equations. The effect is the same as introducing a weighting factor. So the two methods are the same modulo this weighting factor. If need arises for a non arbitrary weight factor, it can always be added later.

It is generally accepted from the above that a good choice for the templates g_q is a set of lowpass filters of different bandwidth that form a pyramid. An alternative is to have g_q's to be shift operators. A shift operator is a template that is zero except for one pixel (other than the $[0,0]$). Convolution with such an operator is equivalent to shifting. Then the two approaches can be made exactly the same.

As mentioned above, the typical hierarchical method [1] works by starting from the coarsest resolution to the finer, using at each level the result of the previous level as a guess. One can start by combining a few of the coarsest scales in one equation and moving to the finer by combining finer and finer scales. An even more general scenario is to combine the scales with different weights. At the beginning the coarsest levels have the highest weights and as the computation advances the finer levels get more weight.

If we define

$$I_x^{(g_q)} = \sum_{m,n} I_x[m,n] g_q[i-m, j-n]$$

$$I_y^{(g_q)} = \sum_{m,n} I_y[m,n] g_q[i-m, j-n]$$

$$I_t^{(g_q)} = \sum_{m,n} I_t[m,n] g_q[i-m, j-n]$$

then

$$err[i,j] = \sum_q \left\{ I_{err}^{(g_q)}[i,j] \right\}^2$$

where

$$I_{err}^{(g_q)}[i,j] = I_x^{(g_q)}[i,j]u[i,j] + I_y^{(g_q)}[i,j]v[i,j] + I_t^{(g_q)}[i,j]$$

We can sum up all the $err[i,j]$'s and try to minimize the sum. But using the same line of thought as before we can attempt to weight parts of the spectrum of $I_{err}^{(g_q)}$ differently by using a filter W with autocorrelation R_w. And of course it makes sense to choose different filters W_q for different g_q's. So we finally get the sum of square errors

$$SSE = \sum_q \sum_{i,j} \left(W_q \otimes I_{err}^{(g_q)} \right)_{i,j}^2$$

and from this we get as before

$$\rho_u = \sum_q I_x^{(g_q)} \; R_{wq} \otimes I_{err}^{(g_q)}$$

$$\rho_v = \sum_q I_y^{(g_q)} \; R_{wq} \otimes I_{err}^{(g_q)}$$

and the preconditioners are

$$B_1 = \sum_q I_x^{(g_q)^2}[k,l] R_{wq}[0,0]$$

$$B_2 = \sum_q I_x^{(g_q)}[k,l] I_y^{(g_q)}[k,l] R_{wq}[0,0]$$

$$B_3 = B_2$$

$$B_4 = \sum_q I_y^{(g_q)^2}[k,l] R_{wq}[0,0]$$

Simple inspection shows that unless the filters g_q and R_{wq} are poorly chosen, the equations are independent. The determinant $B_1^2 - B_2^2$ can be used as confidence measure.

5.3. Choosing filters

We have to select two sets of filters. The g_q's and the R_{wq}'s where $q = 1..q_{\max}$. As explained above, if an image dilates or rotates then its Fourier transform undergoes similar transformations. If we choose the g_q's to have a Fourier transform that is not flat, then as the

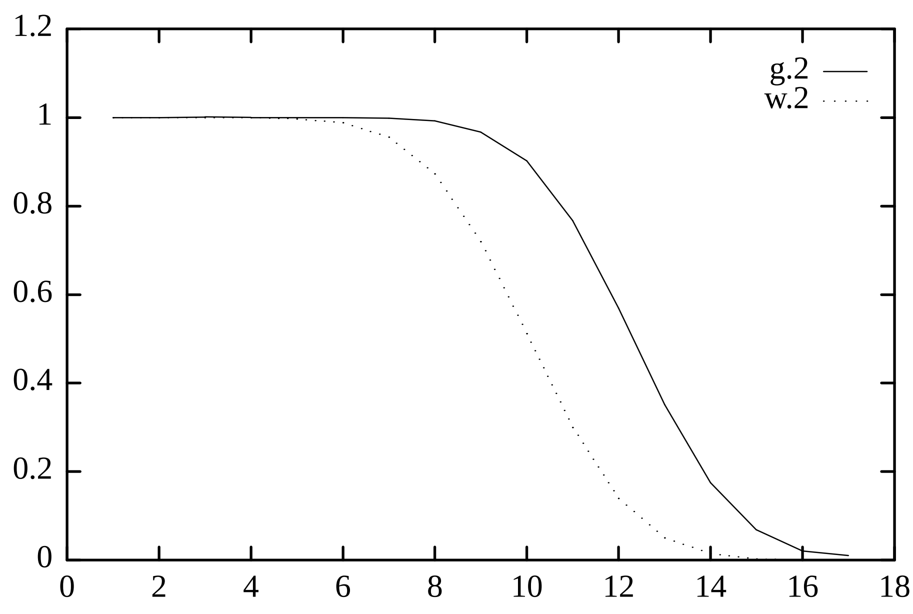


Figure 5.1 The amplitude of the Fourier transform of the g and w filters. The filter w should cut the frequencies where the response of g changes.

frequencies that compose the image move then they might go from a location of high response in the spectrum to a location of low response. This would create a considerable residual error. It would be best if the spectrum of the g_q's is constant α in a region and zero outside the region. Then a good matching filter R_{wq} would be one that is zero in the area around the border of the region. This way frequencies that may cross the border will be cut off.

6. Experiments

The algorithm was implemented on MediaMath [12] and tested on real and synthetic images. Jacobi iteration was used as a linear system solver. Two levels of resolution were used and a smoothing term with $\lambda = 2.0$.

6.1. Synthetic images

Two sets of synthetic images were used. One consisted of six blobs that were moving independently. The other of two cosines in polar coordinates that were rotating.

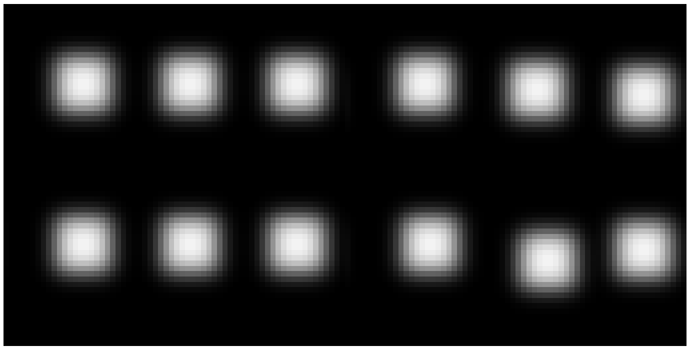


Figure 6.1 The 64×64 synthetic image of the blobs.

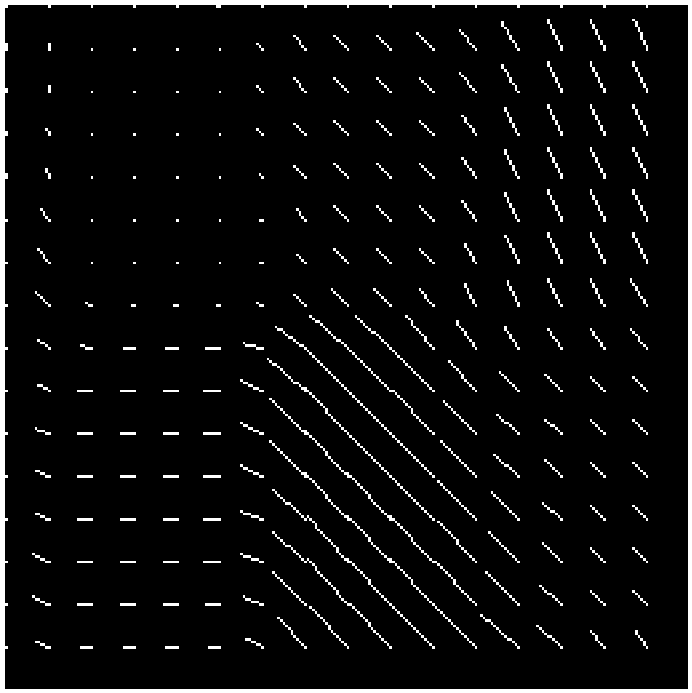


Figure 6.1 The optical flow of the moving blobs.

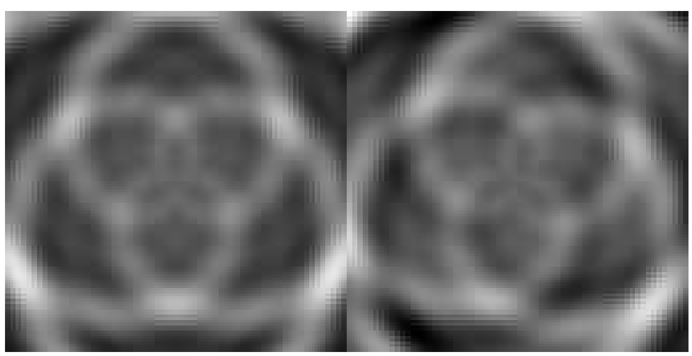


Figure 6.1 The 64×64 synthetic image of the polar cosines.

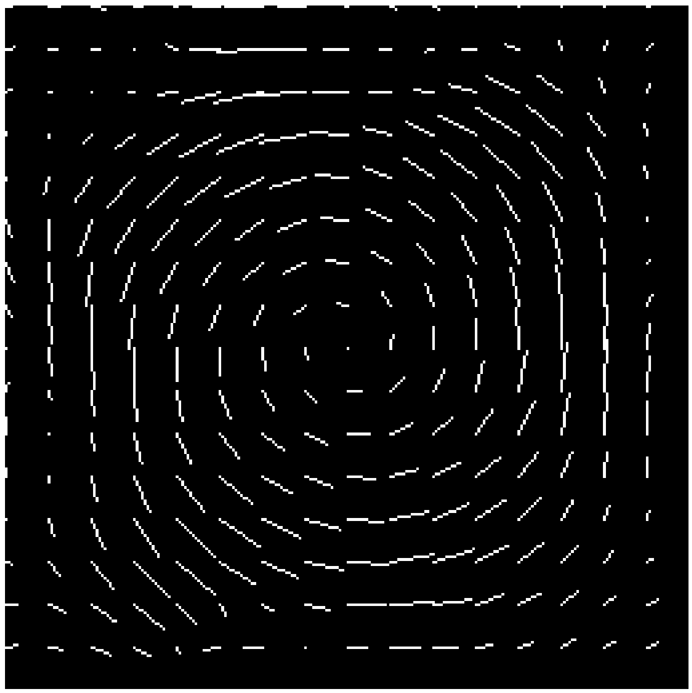


Figure 6.1 The flow of the polar cosines.

6.2. Real images

The real images used were the “NASA sequence” by Banavar Sridar that was made available for the Workshop on Motion 1991. The other image was provided by David Fleet and Leif Haglund.

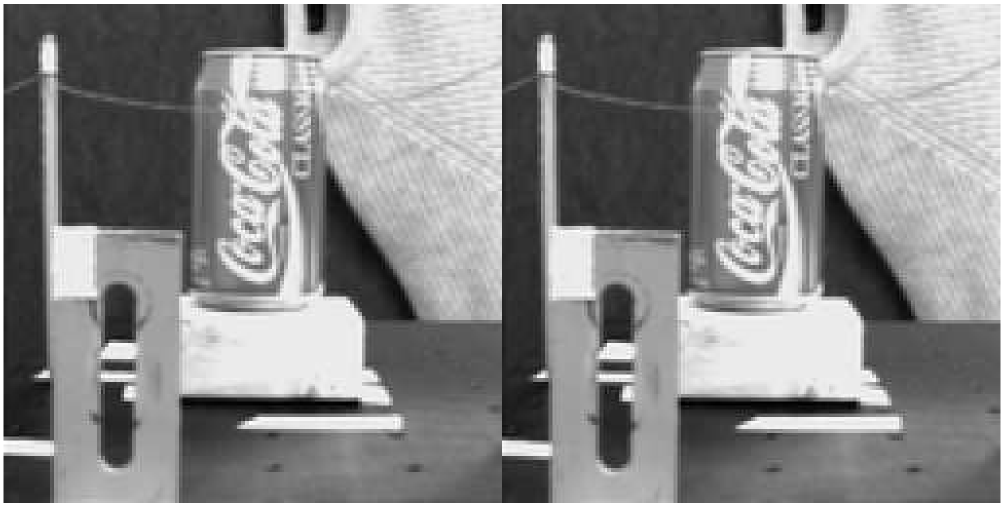


Figure 6.1 The first and third images of the NASA sequense shrunk and trimmed to 128×128.

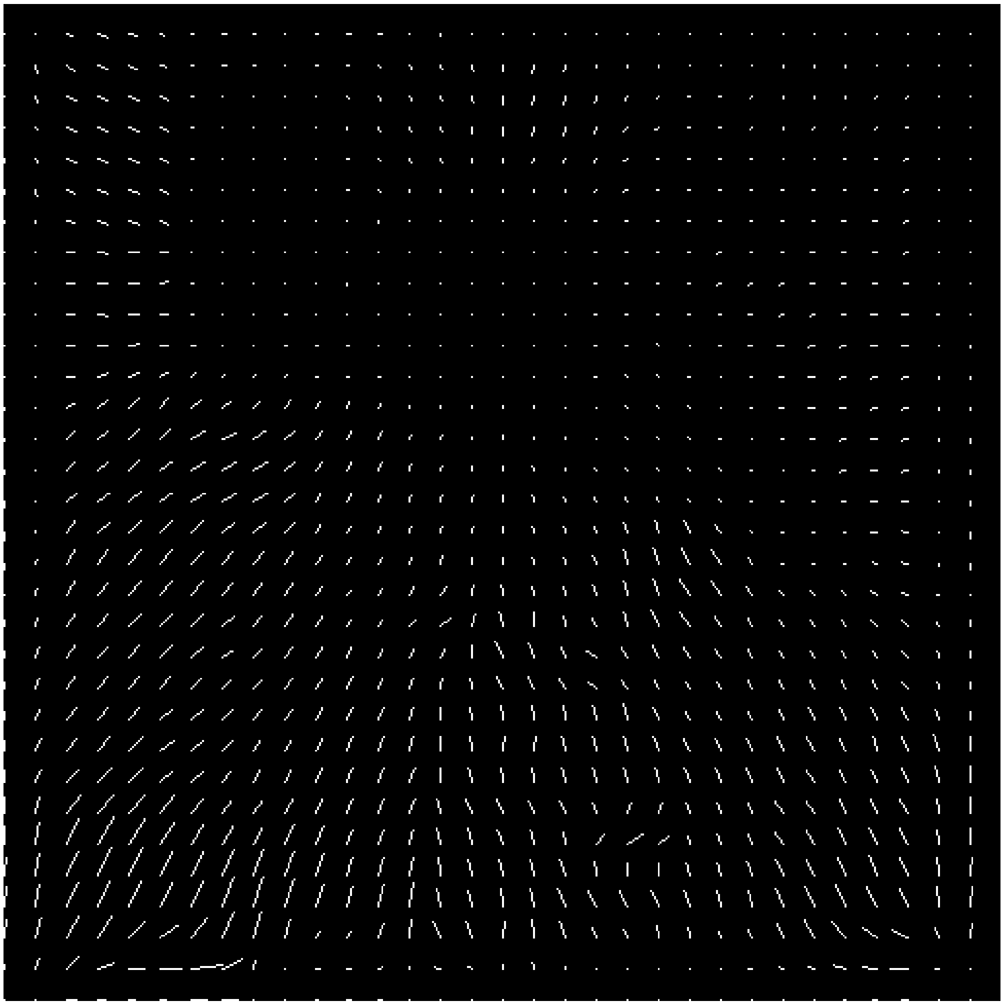


Figure 6.1 The optical flow of the above sequence.

Figure 6.1 The diverging tree sequense trimmed to 128×128. These are the 12^{th} and 15^{th} images from the sequense.

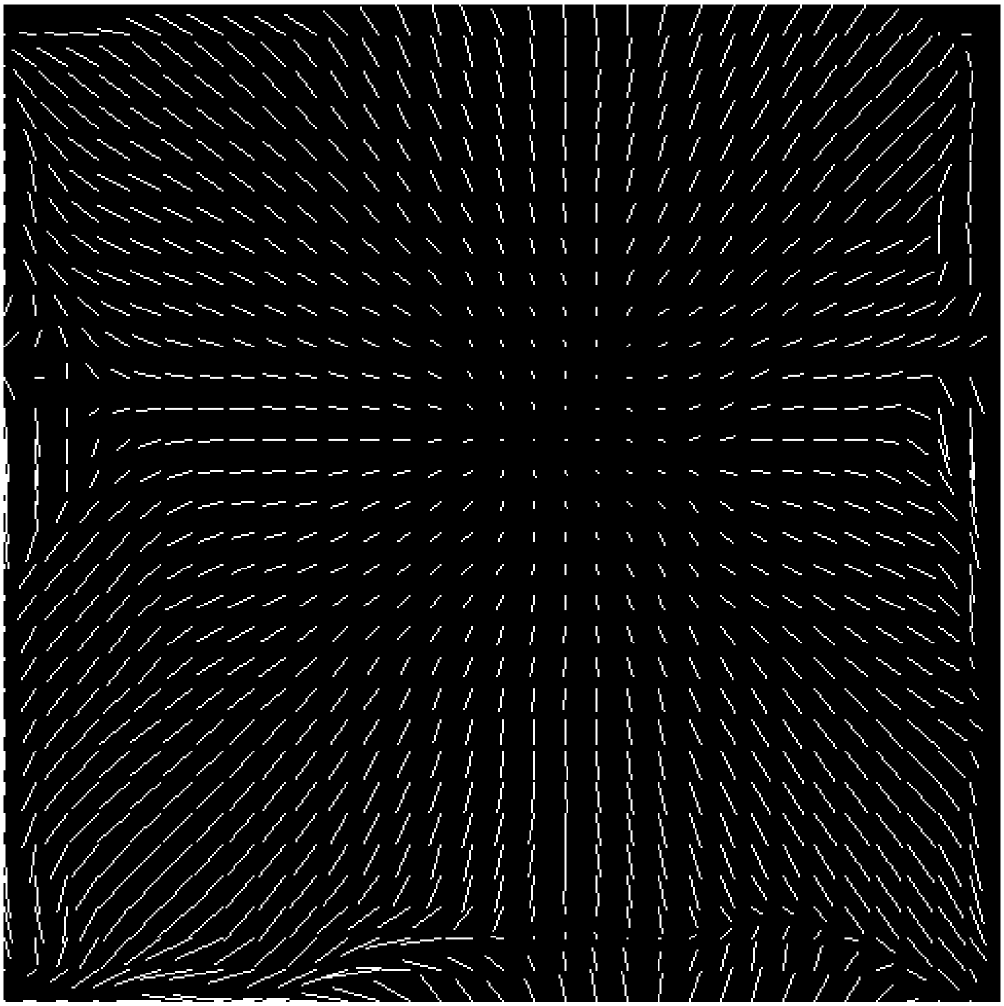


Figure 6.1 The optical flow of the diverging tree sequense.

7. Conclusions

The value of proper weighting is hard to overestimate. In this paper we presented a method that can incorporate a weighting to counter the interdependance among pixels. This technique can be applied in conjunction with several other techniques.

It is obvious that the attenuation of the “bad” areas of the spectrum cannot done by simply prefiltering the image. The prefiltering would simply strengthen the effects of spectrum warping.

The method can be improved by studying the exact quantitative nature of the correlation the pixels to design filters that are accurately tuned for flow estimation.

References

1. P. Anandan, “A Computational Framework and an Algorithm for the Measurement of Visual Motion,” *Intl’ J. of Computer Vision* **2** pp. 283-310 (1989).

2. J. L. Barron, D. J. Fleet, and S. S. Beauchemin, *Performance of Optical Flow Techniques,* RPL-TR-9107, Robotics and Perception Lab, Queen’s University (Jul2 1993).

3. J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg, “Transparent-motion analysis,” *ECCV,* pp. 566-569 (1990).

4. D. J. Fleet and A. D. Jepson, “Computation of component image velocity local phase information,” *Intl’ Journal of Computer Vision* **5** pp. 77-104 (1990).

5. D. J. Fleet and K. Langley, “Toward Real Time Optical Flow,” *Vision Interface,* pp. 116-124 (1993).

6. D. J. Fleet and K. Langley, *Recursive Filters for Optical Flow,* RPL-TR-9308, Robotics and Perception Lab, Queen’s University (May 1993).

7. B. K. P. Horn and B. G. Schunck, “Determining Optical Flow,” *Artificial Intelligence* **17** pp. 185-204 (1981).

8. A. Jepson and M. Black, “Mixture models for optical flow computation,” *CVPR,* pp. 760-761 (1993).

9. B. Lucas, *Generalized Image Matching by the Method of Differences,* PhD Dissertation, Dept. of Computer Science, Carnegie Mellon University (1984).

10. H. H. Nagel, “Displacement Vectors Derived from Second order Intensity Variations in Image Sequences,” *CVGIP* **21** pp. 85 - 117 (1983).

11. A. Singh, *Optic Flow Computation: A Unified Perspective*, IEEE Computer Society Press (1991).

12. Minas Spetsakis, “MediaMath: A reasearch environment for vision research,” *Vision Interface*, (1994).

13. A. Verri and T. Poggio, “Against quantitative optical flow,” *ICCV*, pp. 171-180 (1987).

14. D. Weinshall, “Direct computation of Qualitative 3D shape and motion invariants,” *ICCV*, pp. 230-237 (1990).

15. J. Weng, “Image matching using the windowed Fourier phase,” *IJCV* **11(3)** pp. 211-236 (1993).