Spatial Filtering Requirements for Gradient-Based Optical Flow Measurement

W.J. Christmas
Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford GU2 5XH, UK
w.christmas@ee.surrey.ac.uk

Abstract

When using a gradient-based method to determine the optical flow field for an image sequence, it is generally appreciated that some spatial pre-filtering of the images is usually needed, particularly for large motion values. However the characteristics of the filter are not generally given.

In this paper we analyse the motion measurement from the point of view of sampling theory. We show how an aliasing problem can arise due to undersampling in the temporal domain, and how this problem can be alleviated by appropriate post-sampling spatial filtering. We also demonstrate a connection between this filtering and the methods used to generate the spatial and temporal image intensity gradients.

1 Introduction

In all forms of optic flow measurement, if accurate measurements are required, we find that some form of initial estimate of the likely range of motion values is needed; thus there is an inherent bootstrap problem. For block-based methods, either optimisation techniques are used [2, 7], or a full search is carried out (*e.g.* using phase correlation techniques[9]). In the former case, some form of pre-filtering is generally used in order to remove local minima, whereas in the latter case, some maximum size has to be set for the region that is to be searched.

There is an equivalent problem for gradient-based methods; thus for example Tekalp[8] notes (p. 85) that "Spatial and temporal pre-smoothing of video with Gaussian kernels usually helps gradient estimation"; Weng *et al*. [10] specify a 3×3 filter, but do not specify the weights or justify the choice of mask size. This begs the questions: (a) what type of kernel should we choose and (b) how much filtering (how wide a kernel should we use)?

One way of looking at these problems is from the point of view of sampling theory. From this viewpoint, we find that image motion can result in temporal aliasing (briefly discussed in [1]), in particular when estimating the temporal image gradient. In other words, the temporal sampling frequency (the frame rate) is too low. However, we also find that the temporal aliasing problem can be mitigated by an appropriate level of spatial filtering.

BMVC 1998 doi:10.5244/C.12.19

A subsidiary issue is the method used for estimating the image intensity gradients. Generally a simple forward or central difference is used: these have frequency responses that are far from that of an ideal generator. We show that the use of a more sophisticated estimator can improve the accuracy of the result.

In the next section, we show why image motion causes temporal aliasing, and how to avoid the problem by spatial filtering. This is followed in Section 3 by a discussion on the calculation of derivatives. In Section 4 we show how both the amount of filtering and the accuracy of the calculation of the derivatives affect the accuracy of the motion estimation.

2 Temporal aliasing and how to avoid it

In this section we discuss two points. Firstly, in an image sequence in which there is motion of large magnitude, we find that there is a likelihood of temporal aliasing, whose severity increases with the magnitude of the motion. This causes a problem for the estimation of the temporal derivative. Secondly, the effects of this aliasing can be removed by appropriate spatial low-pass filtering.

To make the analysis simpler, we consider an image sequence with one spatial dimension. However it can easily be generalised to a two-dimensional image sequence: the single spatial axis of the analysis can be thought of as being aligned in the direction of the spatial intensity gradient ∇h. Since there is no intensity variation in the perpendicular direction (to a first-order approximation), we ignore the effects of temporal aliasing in this direction.

Sampling in the spatial domain does not affect the arguments that follow, except insofar that spatial sampling limits the extent of the spatial frequency content.

2.1 Temporal aliasing due to motion

Consider a one-dimensional time-varying image $h(x,t)$ of a scene under constant illumination moving with (unknown) velocity v. Thus for this special case we can say that

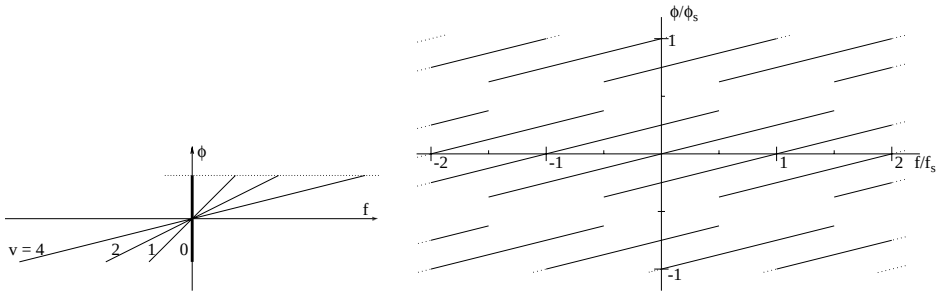
$$h(x,t) = h(x - vt)$$

Hence if $H(\phi)$ is the 1-D transform of the stationary image $h(x)$, the Fourier transform $H(\phi, f)$ of $h(x,t)$ can be written as

$$H(\phi, f) = H(\phi)\delta(v\phi + f)$$

where $\delta(v\phi + f)$ is the 1-D Dirac delta function embedded in two dimensional frequency space. Thus $H(\phi, f)$is non-zero only on the line $f = -v\phi$. This is illustrated in Fig. 1(a), which shows where the spectral components will lie in the $\phi - f$ plane for different velocities. Note that each spectrum excurses by the same amount in the ϕ-direction, whereas the excursions in the f-direction increase in proportion to the velocity v.

If the image is sampled in the time and spatial domains with sampling frequencies f_s and ϕ_s respectively, the spectrum of $H(\phi, f)$ in Fig. 1(a) will be replicated in the f- and ϕ-directions at intervals of f_s and ϕ_s respectively. This is illustrated for a single velocity in Fig. 1(b), for a motion of 4 pixels / frame, and with a spatial spectrum $H(\phi)$ extending to $\pm 3/8\phi_s$.



(a) different image velocities (b) after temporal and spatial sampling, for $v = 4$

Figure 1: Spectra of image with uniform motion

The temporal spectrum at a particular position x_0 is given by the 1-D Fourier transform $H_{x_0}(f)$ of $h(x_0, t)$:

$$H_{x_0}(f) = \frac{1}{v} e^{i2\pi f x_0 / v} H\left(-\frac{f}{v}\right) \tag{1}$$

This corresponds to a projection of the 2-D spectra onto the f-axis, from which we can see that the contributions from the various temporal replications will overlap; *i.e.* temporal aliasing[1] will occur. Note that in the converse situation, *i.e.* for the spatial spectrum at a particular time instant, there is no equivalent problem because the effect of image motion is to "stretch" the spectrum in the f-direction only.

The temporal aliasing will potentially affect any temporal filtering operations, such as the calculation of the temporal image gradient $\frac{\partial h}{\partial t}$. The problem can happen for values of v greater than 1 pixel/frame, and will clearly get worse with increasing v, since from (1) the spectrum is "stretched" in the temporal direction by an factor of v.

2.2 Spatial filtering can remove temporal aliasing

In conventional time-domain signal processing, temporal aliasing is removed by filtering the signal with a suitable analogue low-pass filter, with a cutoff frequency of $f_s/2$, before it is sampled. The cutoff frequency of this filter is such that no overlapping of the replicated spectra can occur. In the case of image sequences, temporal filtering before temporal sampling is problematic, since temporal sampling usually happens at the moment of acquisition.[2] However, we can see from Fig. 1(b) that the temporal aliasing can nevertheless be removed by an appropriate *spatial* filter, shown in Fig. 2. When the replicated spectra are projected onto the f-axis, we can see that they no longer overlap.

[1] Some authors (*e.g.* [4]) use the term "aliasing" to refer to the spectral replication resulting from sampling, rather than to the spectral interference that can result. Although the former is a more logical usage, the latter is more widespread, and is used here.

[2] Perhaps we should qualify this by noting that, for many cameras, some temporal filtering is effected by the temporal aperture of the camera. This typically has a frequency response of $sin(\pi f/f_s)/(\pi f/f_s)$, where f_s is the frame rate. This response is far from ideal, but is at least acting in the right spatial direction.

The required spatial cutoff frequency to achieve this must be less than $\phi_s/2v$. Thus, for "perfect" filtering, we need some estimate of the maximum likely image motion.

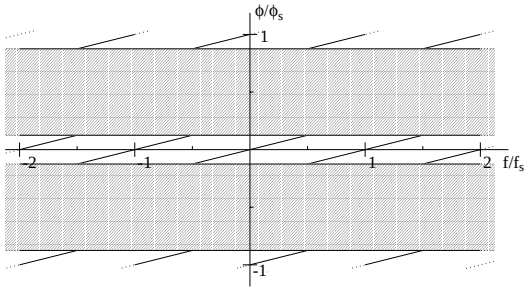


Figure 2: Spectrum of signal after low-pass spatial filtering

The penalty for such filtering can be severe. Since the motion measurement is performed on the filtered image, inaccuracies can be expected near object boundaries. Thus some iterative approach is usually required, in which the motion is compensated for after each stage, so that the filtering can be made progressively less severe.

2.3 Practical filtering strategies

From the foregoing, the ideal spatial filter would be an ideal 1-D low-pass filter with a cutoff frequency of $\phi_s/2v$, acting in the direction of motion. In practice, of course we know neither the magnitude or direction of the motion in advance. We might therefore assume some upper limit v_u for the likely motion magnitude, and use a 2-D isotropic filter, with cutoff frequency of $\phi_s/2v_u$. Some practical requirements are:

- The response should be as close to zero as possible throughout the stop-band; however, in the pass-band, the accuracy of the response is less critical.
- The filter should be efficient to implement, which implies that (a) the filter should be separable into 1-D components, and (b) the number of coefficients for the 1-D filter should not be too large.

Anandan *et al.* [1] noted that the aliasing problem is reduced if a hierarchical motion measurement method is used, since spatial subsampling in effect reduces the image motion magnitude by the subsampling factor.

3 The calculation of derivatives in sampled image sequences

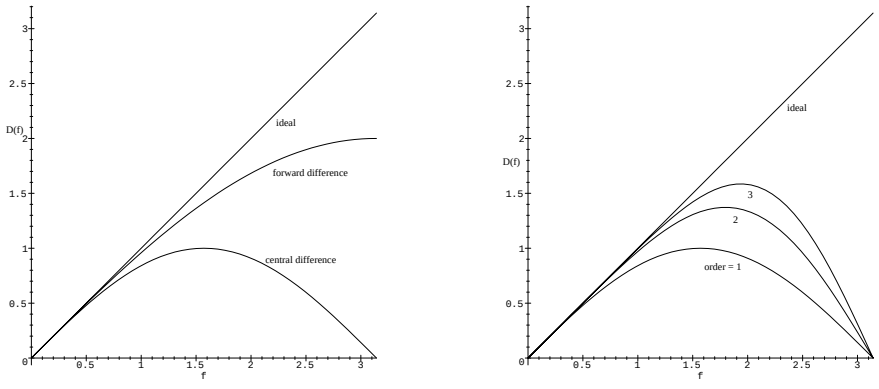
Consider a 1-D signal $h(t)$. If it is to be sampled at a frequency f_s, it should have no frequency components above $f_s/2$. Thus the frequency response of the ideal differentiator, $D(f)$, is given by

$$D(f) = i2\pi f, \quad |f| < f_s/2$$

By comparison, the forward and central difference methods that are usually used in image processing have frequency responses $D_F(f)$ and $D_C(f)$ respectively:

$$D_F(f) = i2f_s e^{i\pi f/f_s} sin(\pi f/f_s), \quad D_C(f) = if_s sin(\pi 2f/f_s)$$

The magnitudes of these responses are compared in Fig. 3(a). We can see that the central difference departs significantly from the ideal response at higher frequencies. While the forward difference amplitude response is more accurate, there is a phase error which can be problematic; we therefore used central difference schemes in our experiments.



(a) different schemes (b) different orders

Figure 3: Comparison of frequency responses of difference schemes

The signal processing literature has standard techniques for improving the performance of derivative estimators, by the addition of more terms[5]. These generally aim to optimise the frequency response over a fixed frequency range. In our case, we note (from Fig 2) that:

- the required temporal frequency range can extend over the whole spectrum;
- in the spatial direction it can extend up to the anti-alias filter cutoff frequency $\phi_s/2v_u$ (and is therefore dependent on the unknown motion), and
- insofar as one can generalise, in typical images the spatial frequency content falls off roughly as $1/\phi$.

We used a single design for temporal and spatial directions that optimises the fit of the estimator at low frequencies. Specifically, for an nth order estimator, the first n derivatives of the estimator frequency response are equated to those of the ideal response. The coefficients for central difference estimators up to 3rd order are:

order	c_{-3}	c_{-2}	c_{-1}	c_0	c_1	c_2	c_3
1			-1	0	1		
2		$\frac{1}{12}$	$-\frac{2}{3}$	0	$\frac{2}{3}$	$-\frac{1}{12}$	
3	$-\frac{1}{60}$	$\frac{3}{20}$	$-\frac{3}{4}$	0	$\frac{3}{4}$	$-\frac{3}{20}$	$\frac{1}{60}$

The frequency responses are compared in Fig. 3(b), from which we can see that:

- on the one hand, all of these difference schemes will underestimate the derivative;
- on the other hand, the more severe the anti-alias filter, the smaller this error will be, since the errors increase monotonically with frequency.

Note that we could have used the same design technique as was used for the spatial low-pass filter in Section 2. However, this requires a different optimisation for each version of the spatial filter cutoff frequency $\phi_s/2v_c$. Also as the cutoff frequency tends to zero, the coefficients of this method tend towards those of our method.

4 Experiments

In Sections 2 and 3 we discussed two sources of error, from temporal aliasing and inadequate differentiator design, together with claims for their remedies. In this section we show the results of some experiments to test these claims. For this we used a simple optical flow algorithm which we briefly describe.

Assume that some region of the image $h(\mathbf{x}, t)$ is moving, under constant illumination, with uniform velocity $\mathbf{v}$. Thus in the coordinates of the region, $\frac{dh}{dt} = 0$. Applying the chain rule for partial differentiation, we can express this in terms of the image coordinates:

$$\mathbf{v}.\nabla h + \frac{\partial h}{\partial t} = 0 \tag{2}$$

In order to avoid the aperture problem[11], and to reduce the effect of noise, we solve a set of such equations over a small patch of image pixels. This is then merely a standard linear regression problem of fitting the values of $\frac{\partial h}{\partial t}$ to the vectors ∇h, subject to an unknown vector of proportionality $-\mathbf{v}$. For simplicity, we use a least mean squared regression fit. Thus if there are N pixels in the patch, and we denote by $\left[\frac{\partial h}{\partial t}\right]$ and $[\nabla h]$ the column vector and $N \times 2$ matrix formed by stacking the values over the patch of $\frac{\partial h}{\partial t}$ and ∇h respectively, the solution [1, 3] is given by:

$$\hat{\mathbf{v}} = \left([\nabla h]^T [\nabla h]\right)^{-1} [\nabla h]^T \left[\frac{\partial h}{\partial t}\right] \tag{3}$$

The matrix $[\nabla h]^T [\nabla h]$ was inverted using eigenvalue decomposition; when the matrix was ill-conditioned, the eigenvalues were progressively adjusted to bias the result towards zero motion.

4.1 Experimenting with a synthetic sequence

The image used was generated from a set of independently randomly-generated pixels, uniformly distributed. This image is chosen as it has a known (flat) spectral content, and contains a significant amount of high-frequency information: the effects of both the anti-alias filter and the use of better derivative estimators show up more at higher spatio-temporal frequencies. The motion was synthesised by shifting the image by 4 pixels per frame in a horizontal direction. Noise with a uniform distribution of $\pm 5/256$ of the image pixel value range was added.

We initially used a 1-D equi-ripple design[6], with a maximum pass-band ripple of 3dB and maximum stop-band ripple of -100dB, applied to the image successively in x- and y- directions. The transition region width was equal to the pass-band width; *i.e.* for an estimated maximum motion magnitude of v_u, the transition region extended from $\phi_s/4v_u$ to $\phi_s/2v_u$.

We compared the results of using (3) with the ground truth. The patch is square, with $N = (2v_u + 1)^2$ pixels (or 9 pixels when no filtering is used). We examine the mean value, to reveal any bias in the method, and the standard deviation of the remaining error. Table 1 shows the mean of the estimated velocity in the horizontal direction, together with

$v_u =$		none	2	4	6	8
estimator	1	10^{-2} [0.58]	0.9 [1.0]	3.0 [0.45]	3.5 [0.31]	3.6 [0.29]
order:	2	10^{-2} [0.59]	1.2 [1.3]	3.6 [0.33]	3.9 [0.21]	4.0 [0.32]
	3	10^{-1} [0.59]	1.4 [1.5]	3.8 [0.26]	4.0 [0.23]	4.0 [0.32]

Table 1: Mean horizontal velocity estimate $\overline{\hat{v_x}}$ and [s.d. of error]

the standard deviation of the error; the mean velocity estimate in the vertical direction was always small (less than 0.1 pixel/frame), as might be expected. We interpret the trends in the table as follows:

Estimator bias: For additive noise, under appropriate assumptions of statistical independence, we can show that (3) is a biased estimator for $\mathbf{v}$; in particular, $\hat{\mathbf{v}} \approx \mathbf{v}/(1 + 1/r_{sn})$, where r_{sn} is the signal-to-noise power ratio. Thus if there are alias components present that behave as random noise, we would expect the values of $\hat{\mathbf{v}}$ to be progressively biased towards zero as the filter cutoff frequency $\phi_s/2v_u$ is increased beyond $\phi_s/2|\mathbf{v}|$. This is certainly borne out in Table 1: the bias increases dramatically for $v_u < 4$.

We also remember that the differential estimators underestimate the correct value, particularly at higher frequencies. For $|\mathbf{v}| > 1$ this will affect the temporal derivatives more than the spatial ones (Fig.1(b)). We might therefore expect from (3) that increasing the differentiator estimator order should remove some of the bias; this is most noticeable in the table at $v_u = 4$. Decreasing the filter cutoff frequency should have a similar effect (quite apart from aliasing considerations), as this will further remove the contribution of the higher frequency parts of the differentiator estimator response. In the table, we can see that there is still some significant improvement as v_u increases beyond the ground truth motion value of 4.

Standard deviation: We would also expect the standard deviation to decrease with better filtering, reflecting a better signal-to-noise ratio. Table 1 in general bears this out. The one exception is for the case of no filtering, where the standard deviation is less than the case of $v_u = 2$. This is presumably because the severe bias of $\hat{\mathbf{v}}$ towards zero will scale the standard deviation similarly.

4.2 Comparison with other filters

We next compare the effect of using the low-pass filter suggested by Section 2.3 with that of two other frequently-used filter types: a Gaussian filter, and a simple averaging filter.

4.2.1 Gaussian filter

A 1-D Gaussian filter was applied in two orthogonal directions; the filter impulse response $g(x)$ is given by:

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{x^2}{2\sigma^2}\right)$$

A filter of order 3σ was used, with a raised-cosine window. Table 2 shows the effect of varying the filter width parameter σ.

$\sigma =$		2	4	6	8	12	16
estimator	1	1.2 [1.3]	2.8 [0.67]	3.2 [0.39]	3.4 [0.28]	3.5 [0.20]	3.5 [0.16]
order:	2	1.5 [1.5]	3.3 [0.58]	3.7 [0.29]	3.8 [0.23]	3.9 [0.17]	3.9 [0.14]
	3	1.6 [1.7]	3.6 [0.52]	3.9 [0.24]	4.0 [0.20]	4.0 [0.16]	4.0 [0.13]

Table 2: As Table 1, but using Gaussian filter

It appears that similar results are obtained using the Gaussian design. However, generous filter orders were used in both cases, so more experimentation with the filter order is needed, particularly for real-time applications.

4.2.2 Averaging filter

We tried simple averaging filters with a range of widths from 2 to 16 pixels. None of them gave a mean velocity estimate of more than 25% of the correct value.

4.3 Real image sequences

What of sequences of real images? The synthetic sequence used in the previous section was chosen for its flat frequency response, since we were investigating Fourier domain effects. Real images on the other hand tend to have spectra weighted towards lower frequencies, so that some of the required filtering has in effect already been done. On the other hand the spectra are of course also very variable, so that we cannot predict *how much* filtering needs to be done. Another difficulty with the use of real image sequences is the lack of ground truth motion vectors. It is not sufficient to use a single image and artificially displace it: the effects of temporal camera aperture (footnote, Section 2.2) would then be unrealistic. We therefore make only qualitative comparisons between the effects of different filters.

The sequence used was from the "foreman" image sequence (Fig. 4). The motion results from camera movement; thus it is locally uniform, which is appropriate for our simple algorithm. The maximum motion component was estimated to be around 8 pixels per frame. The algorithm was run using the three filter types, and with a 2nd-order differencing scheme. The best results were obtained with the equi-ripple filter. The results for $v_u = 0$, 4, 8 and 12 are shown in Fig. 5; the arrows indicate the motion vectors, and the circles the standard deviation of the regression fit over the patch. The symbols are plotted every 10 pixels; the arrows are shown to scale. From these results we can see that, provided that $v_u > |\mathbf{v}|$, good measurements can be made. The results using corresponding Gaussian filters were similar, while those using simple averaging filters were substantially worse.

Figure 4: Sample image from sequence

5 Conclusions

From the above results we reach the following conclusions:

- A significant amount of motion can create temporal aliasing, which can affect the accuracy of motion measurement (as is well known).
- This problem can be alleviated by spatial filtering. The filter width depends on the maximum motion magnitude likely to be encountered. Although the specified filter shape is expensive to realise, in practice simpler filters may be adequate.
- Further improvements to accuracy can also be made by using higher-order estimates of the image intensity derivatives.

We can also conclude that hierarchical motion measurement schemes, in which the images are progressively subsampled, could also alleviate the aliasing problem, provided that (a) there are enough levels in the hierarchy to ensure that the motion in the final subsampled image is less than about 1 pixel per frame, and (b) the image is filtered appropriately each time it is subsampled.

Acknowledgements

This work was supported by EPSRC Grant GR/K42276.

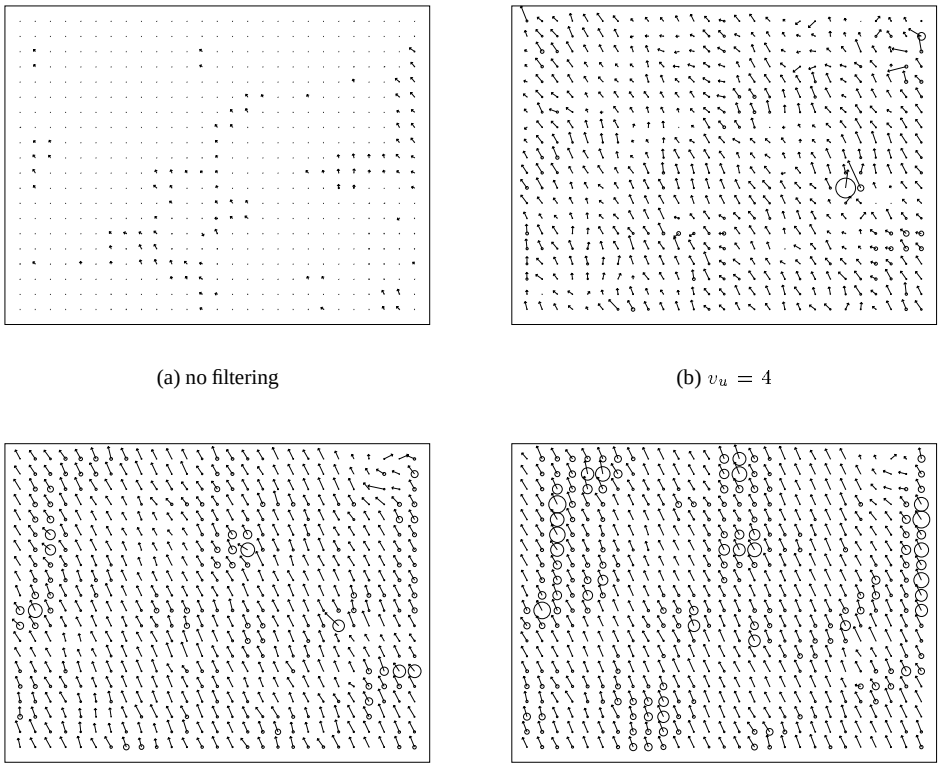
References

[1] P. Anandan, J.R. Bergen, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In M.I. Sezan and R.L. Lagendijk, editors, *Motion analysis and image sequence processing*. Kluwer Academic Publishers, 1993.

[2] M. Bober and J. Kittler. Robust motion analysis. *Computer Vision and Pattern Recognition*, pages 947–952, 1994.

[3] D.C. Montgomery and E.A. Peck. *Introduction to linear regression analysis*. Wiley, 1982.

[4] A.V Oppenheim and R.W. Schafer. *Digital signal processing*. Prentice-Hall, 1975.



(a) no filtering

(b) $v_u = 4$

(c) $v_u = 8$

(d) $v_u = 12$

Figure 5: Motion vectors for different amounts of filtering, using equi-ripple filter

[5] J.G. Proakis and D.G. Minolakis. *Introduction to digital signal processing*. Macmillan, 1988.

[6] L.R. Rabiner, J.H. McClellan, and T.W. Parks. FIR digital filter design techniques using weighted Chebychev approximations. *Proc. IEEE*, 63(2):595–6102, 1975.

[7] V. Seferidis and M. Ghanbari. Generalised block-matching motion estimation using quad-tree structured spatial decomposition. In *IEE Proc.-Vis. Image Signal Process.*, number 6 in 141, December 1994.

[8] A. Murat Tekalp. *Digital video processing*. Prentice Hall, 1995.

[9] G.A. Thomas. Television motion measurement for DATV and other applications. Technical Report 11, BBC Engineering Research Dept., 1987.

[10] J. Weng, T.S. Huang, and N. Ahuja. *Motion and structure from images*, chapter 2.3.8. Springer-Verlag, 1993.

[11] *ibid.*, chapter 2.2.2.