Optimal Filters For Gradient-Based Motion Estimation

Michael Elad * Patrick teo † Yacov Hel-Or ‡

Abstract

Gradient based approaches for motion estimation (Optical-Flow) refer to those techniques that estimate the motion of an image sequence based on local changes in the image intensities. In order to best evaluate local changes in the intensities specific filters are applied to the image sequence. These filters are typically composed of a spatio-temporal pre-smoothing filter followed by derivative filters. The design of these filters plays an important role in the estimation accuracy. This paper proposes a method for the design of these filters in an optimal manner. Unlike previous approaches that design optimal derivative filters in some sense, the proposed technique defines the optimality directly with respect to the motion estimation goal. One possible result of the suggested approach is a set of image dependent filters, which can be computed prior to the estimation process. An alternative is generic filters, capable of treating the typical (natural) images. Simulations demonstrate the validity of the new design approach.

Keywords: Motion estimation, Optical flow, Pre-smoothing, Gradients computation, Optimal Filters, Constrained Minimization.

*HP Israel Science Center, Technion City, Haifa 32000 Israel, elad@hp.technion.ac.il.
†Stanford University, Computer Science Department, Stanford CA, USA, teo@white.stanford.edu
‡HP Israel Science Center, Technion City, Haifa 32000 Israel, toky@hp.technion.ac.il.

1 Introduction

Estimating motion between two images plays a vital role in many applications [1, 2], and has drawn a lot of research attention during the last two decades or so [1, 2]. There are many ways to approach this problem and indeed many algorithms have been proposed for this task [1, 2, 3]. In a recent work by Barron et. al. [3] a comparative survey of many motion estimation techniques is given. One family of such algorithms which was found to perform well is the family of gradient-based methods, originally proposed by Horn and Schunck [4].

The gradient-based methods emerge from the assumption that the intensity value of a physical point in a scene does not change along the image sequence. This assumption yields directly the *Brightness Change Constraint Equation* (BCCE) [1, 2, 3, 4]. Denoting the intensity values of the image sequence by the function $I(x, y, t)$ of spatial position (x, y) and temporal axis t, the brightness constancy assumption between two consecutive images yields:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \quad . \tag{1}$$

Taking the first order Taylor expansion of the right term around the spatio-temporal point (x, y, t) gives:

$$I(x, y, t) = I(x, y, t) + \frac{\partial I(x, y, t)}{\partial x}\delta x + \frac{\partial I(x, y, t)}{\partial y}\delta y + \frac{\partial I(x, y, t)}{\partial t}\delta t \quad . \tag{2}$$

Defining $(u^x, u^y) = (\delta x/\delta t, \delta y/\delta t)$, this vector specifies the spatial velocity of each spatio-temporal point in the image sequence. Assuming the time difference between consecutive frames to be one ($\delta t = 1$), the above equation gives:

$$I_x(x, y, t)u^x(x, y, t) + I_y(x, y, t)u^y(x, y, t) + I_t(x, y, t) = 0 \tag{3}$$

where I_x, I_y and I_t denote the spatial and temporal derivatives. This equation, which is known as the BCCE, relates the spatial and temporal gradients of an image sequence to the motion vector (u^x, u^y) at each location (x, y, t). Since the above equation forms a single constraint over the two component motion vector, more constraints are required to uniquely recover the motion field. For this purpose, an assumption of smoothness (spatial [1, 2, 3] and/or temporal [1, 5, 6, 7])can be imposed.

One issue that is critical to the implementation of the above BCCE is that image gradients are computed based on sampled information. It is commonly agreed [3, 8] that approximating the spatio-temporal gradients by finite differences produces error in the above equation and subsequently in the estimated motion. One of the major conclusions of Barron [3] is that *"the method of numerical differentiation is very important - differences between first order pixel differencing and higher order central differences are very noticeable"* .

In most implementations spatio-temporal smoothing is typically applied to the image sequence prior to motion estimating [1, 2, 3]. The reasons for this pre-processing stage are the following:

1. Additive noise is amplified during gradient computation. Pre-smoothing attenuates this noise [1, 2, 3].

2. Pre-smoothing approximates the data locally as a tilted plane. This justifies the first order Taylor approximation of the BCCE above [1, 2, 3].

3. Pre-smoothing attenuates spatial and temporal aliasing effects in the image sequence [1, 2, 3].

4. Typical finite gradients are more accurate at low frequencies. Since pre-smoothing reduces the high frequencies, overall gradient estimation is more accurate.

Since pre-smoothing and gradient computation are both Linear and Spatio- Temporal Invariant (LSTI), it is possible to combine them into a single filtering operation. In the most general case, the BCCE can be implemented in the following way:

$$I_x(x,y,t)u^x(x,y,t) + I_y(x,y,t)u^y(x,y,t) + I_t(x,y,t) = \tag{4}$$

$$= \{I * F_1\}(x,y,t)u^x(x,y,t) + \{I * F_2\}(x,y,t)u^y(x,y,t) + \{I * F_3\}(x,y,t) = 0$$

where F_1 F_2 and F_3 are 3-D digital filters of some sort, and $\{A * B\}(x,y,t)$ denotes discrete convolution operation between two 3-D signals.

Several attempts to define or design these filters, together or separately, have been reported in the literature [3, 8]. All these methods treat the above question as a problem of optimally designing gradient operators, overlooking the fact that these gradients are to be used for motion estimation. The question addressed in this paper is that of designing these filters such that they are optimal with respect to the motion estimation goal.

Since 3-D separable filters are easier to implement, it is commonly assumed that F_1, F_2 and F_3 are separable [4, 3, 8]. We adopt this line of reasoning in this paper as well.

The rest of the paper is organized as follows: Section 2 briefly surveys existing approaches for the design of the required filters. Section 3 describes the new approach. In Section 4 several examples demonstrating the new design procedure are given. A summary of the paper and conclusions are given in Section 5.

2 Existing Motion Estimation Filters

2.1 General

The numerical analysis literature contains many methods for approximating gradient filters [2, 9]. Most of the papers describing optical flow estimation through the BCCE apply simple gradient filters like $\frac{1}{2}[-1, 0, 1]$ (e.g. [4]). In many papers the choice of these filters is even not mentioned.

In their original paper [4], Horn and Schunck proposed an approximation of the gradient filter with no pre-smoothing. The gradients were obtained by averaging the first differences over a neigborhood of 2×2 in the image sequence (see figure 1). These gradient estimates refer to a center point of a $2 \times 2 \times 2$ cube (which means that the estimated flow corresponds to points between pixels).

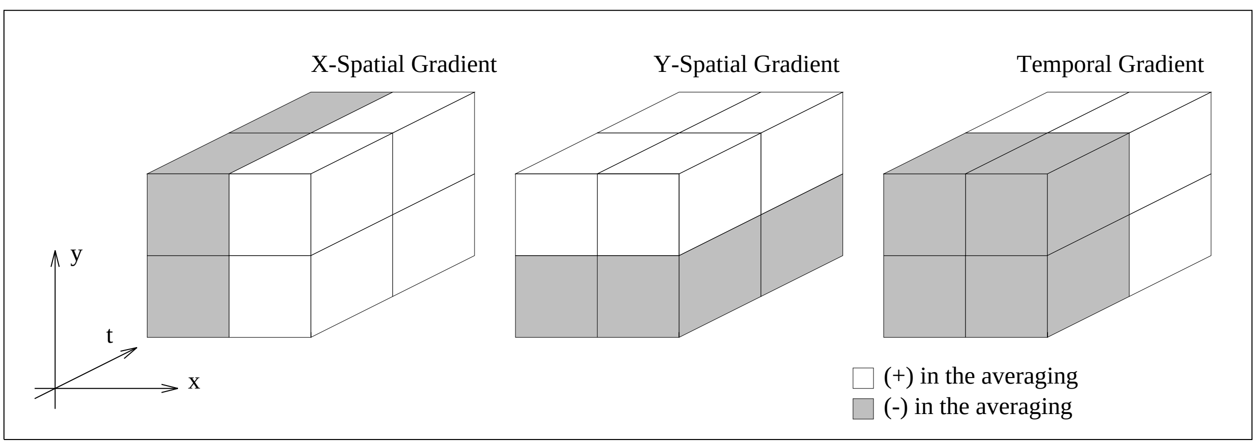


Figure 1: Horn and Schunck proposed filters

No motivation or justification for this choice of gradient estimation is given. According to Barron et. al. [3], these gradient filters are said to be *"relatively crude form of numerical differentiation and can be the source of considerable error"*.

Barron et. al. propose [3] the application of a $5 \times 5 \times 5$ spatio-temporal pre-smoother, constructed using a sampled Gaussian filter with 1.5 variance at each axis. This variance was found empirically to give the best results. The gradient filters proposed by Barron is the 5-tap 1-D filter $\frac{1}{12}[-1, 8, 0, -8, 1]$, which is the result of the design procedure described in [2]. For completeness, the next subsection shortly describes this and related design procedures.

2.2 Optimal Gradient Filters

Let us assume that the required gradient filter is a linear phase Finite Impulse Response (FIR) filter with an odd number of taps. These assumptions are reasonable because:

1. Linear phase filters assure a constant phase delay, which preserves the signal structure [10].
2. Linear phase filters can be implemented by FIR filters [10]. This is, of course, a benefit since FIR filters are easy to implement.
3. A filter with an even number of taps approximates the gradient at the midpoint between two adjacent pixels, whereas a filter with an odd number of taps calculates the gradient at the pixel's center.

This choice of gradient filter implies that its impulse response is antisymmetric [10]. The general form of the filter's impulse response is given by:

$$\{d(k)\}_{k=-L}^{L} = [d_L \; d_{L-1} \; ...... \; d_2 \; d_1 \; 0 \; -d_1 \; -d_2 \; .... \; -d_{L-1} \; -d_L] \quad . \tag{5}$$

Taking the Fourier transform on $d(k)$ yields the following transfer function:

$$D(\theta) = \sum_{k=-L}^{L} d(k)\, exp\{-jk\theta\} = 2j \sum_{k=1}^{L} d_k sin(\theta k) \quad . \tag{6}$$

The goal is to obtain a near-accurate gradient transfer function $D(\theta) = j\theta$. The filter coefficients $\{d(k)\}_{k=1}^{L}$ are designed to meet this requirement as closely as possible. There are several ways to define an optimality criterion in a way that relates these coefficients to the desired transfer function. Among the most popular methods are the Impulse-Response-Truncation (IRT) approach, the windows design approach, the Weighted Least-Squares (WLS) criterion, and the minimax Remez exchange algorithm [10].

The design procedure presented in [2] is different, but resembles the WLS approach. Using the first order Taylor expansion of the function $sin(x)$ in Equation 6, we get for the choice $L = 2$:

$$D(\theta) = 2j\left[d_1 sin(\theta) + d_2 sin(2\theta)\right] = 2j\left[d_1 + 2d_2\right]\theta - \frac{2j}{6}\left[d_1 + 8d_2\right]\theta^3 = j\theta \quad . \tag{7}$$

Thus, setting $2(d_1 + 2d_2) = 1$ and $d_1 + 8d_2 = 0$, the second term vanishes and the approximation holds. The filter obtained is $\frac{1}{12}[-1, 8, 0, -8, 1]$. Since the truncated Taylor expansion error grows as a function of θ, the filter obtained is similar to a weighted least-squares gradient filter, with higher weights at low frequencies. This choice of weights is reasonable if pre-smoothing is applied. Filters with larger number of taps can be obtained using higher orders of the Taylor expansion in a similar manner [2].

The above $1D$ gradient filter is used to produce 3 types of derivatives (x-derivative, y-derivative, and t-derivative) as described in Figure 2. As can be seen, a 3-D $[5 \times 5 \times 5]$ pre-smoothing kernel is first applied to the image sequence. Then, each axis is differentiated separately. The resulting three 3-D kernels F_1, F_2, and F_3 correspond to those found in the BCCE as given in Equation 4.

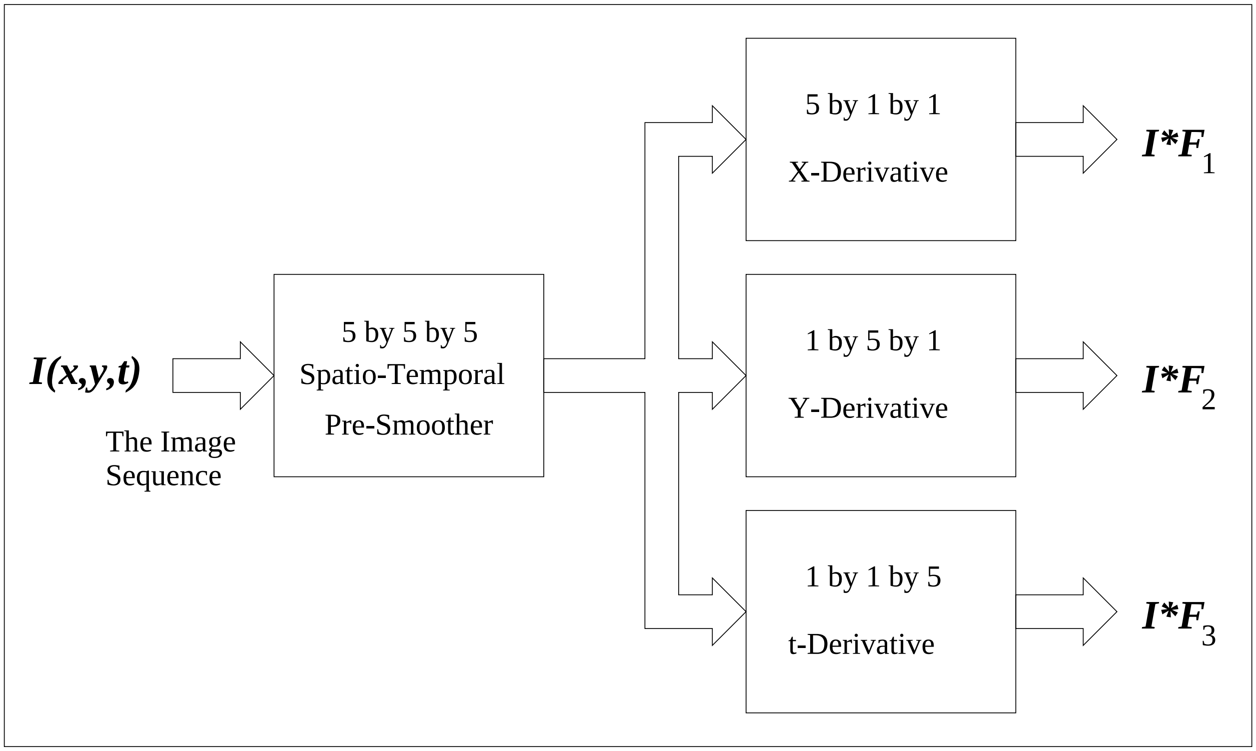
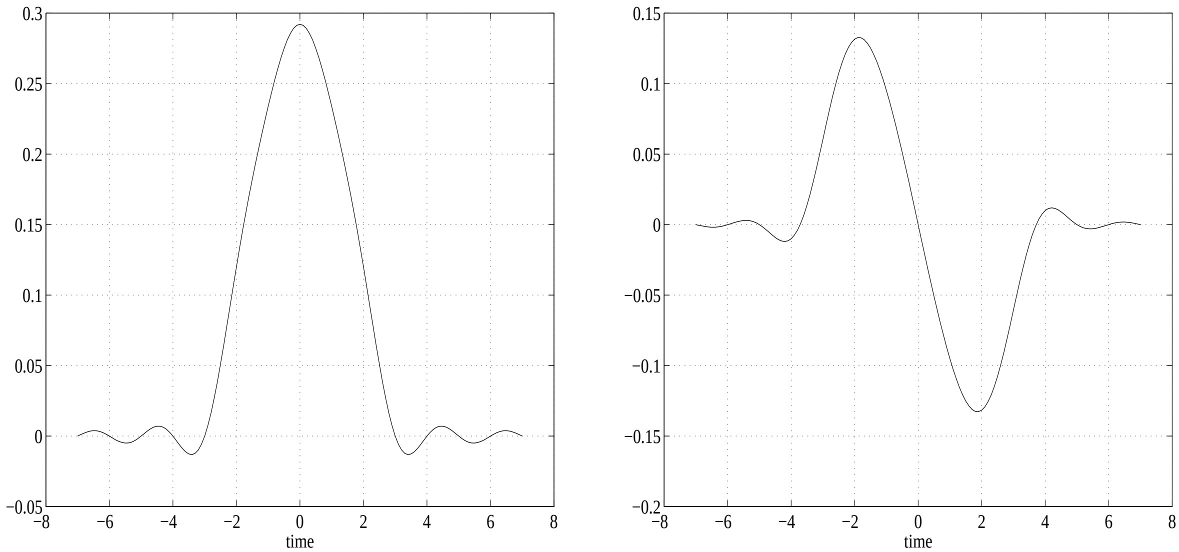


Figure 2: The differentiation scheme as implemented by Barron.

Figure 3 shows the actual 9-taps 1-D filters: pre-smoother $p(x)$ and gradient (combined with the smoother) $d(x) * p(x)$. The pre-smoother is taken to be a sampled Gaussian filter with a variance of 1.5. The gradient filter is the optimal 5-taps filter as described above. The filters are given in their continuous form and interpolated using a Sinc function. Figure 4 depicts the power spectrum of the gradient filter, $|D(\theta)P(\theta)|$, compared to an analytic differentiation of the smoothing filter, $|j\theta \cdot P(\theta)|$. These filters were proposed by Barron et. al. [3] as good candidates for the BCCE approximation. As expected, the error between these two responses is very small for low frequencies but gets higher as the frequency tends to $\pm\pi$.



(a) **(b)**

Figure 3: Pre-smoothing filter $p(x)$ (a) and (b) its derivative $d(x) * p(x)$ (b) as given by applying the optimal gradient suggested by Barron

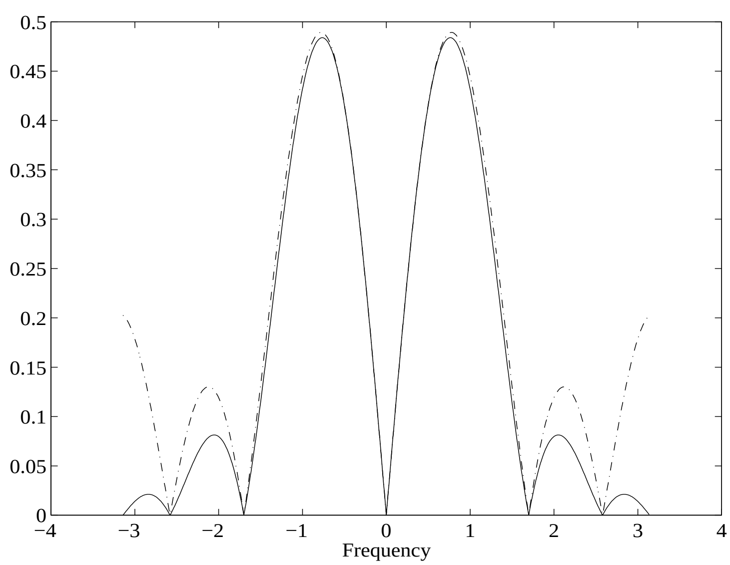


Figure 4: The gradient filters in the frequency domain $|D(\theta)P(\theta)|$ (solid line) and $|j\theta \cdot P(\theta)|$ (dashed line)

2.3 Simoncelli's Filters

In [8] Simoncelli describes a filter design algorithm that takes into consideration several important issues that were traditionally overlooked. These include:

1. **Continuous versus discrete grid** - Most gradient filters are designed on a given discrete grid. However, a given discrete signal is the result of 3-D sampling of the originally continuous image sequence. Thus, the design of discrete gradient filters should be done in the continuous domain. This means that a function $b(x, y, t)$, interpolating the signal to the continuous grid, is required. Simoncelli [8] suggests using the *sinc* function or a more *"gentler function"* such as a Gaussian.

2. **Design of pairs of filters** - Instead of designing a single gradient filter, both the smoothing and the the gradient filters are designed simultaneously. This new idea suggest that a signal $I(x)$ and its derivative $I_x(x)$ are hard to come-by accurately in the discrete domain. Alternatively, we can supply a pair $\{I * p\}(x)$ and $\{I * d\}(x)$ which relate to each other as an original and its gradient in a more accurate form. If we denote a smoothing and a derivating 1-D filter pair in the frequency domain by $P(\theta)$ and $D(\theta)$ respectively, then the error $[j\theta\ P(\theta) - D(\theta)]$ can be minimized in a more accurate manner. For example, high frequencies which are not treated correctly by $D(\theta)$ are attenuated by $P(\theta)$ in order to minimize the above approximation.

3. **Weighting the approximation error** - Since the required filters are designed to work on natural images, it is reasonable to weight the approximation error accordingly. Simoncelli [8] suggested that the design should be done in the frequency domain, and the weight should manifest the $1/\theta$ spectral behavior of typical images.

With the above considerations, Simoncelli [8] suggested the following WLS criterion for the filter design:

$$\begin{aligned} \epsilon^2(\underline{p}, \underline{d}) &= \int_{-\pi}^{\pi} \frac{1}{|\theta|^{0.5}} [j\theta P(\theta) - D(\theta)]^2 d\theta \qquad (8) \\ \text{where} \quad P(\theta) &= \sum_{k=-L}^{L} p\,(k)\, exp\{-jk\theta\} \quad , \\ \text{and} \quad D(\theta) &= \sum_{k=-L}^{L} d\,(k)\, exp\{-jk\theta\} \quad . \end{aligned}$$

The solution is obtained using the Singular Value Decomposition (SVD). The resulting filters are normalized so that $p(k)$ yields a unity DC response. The result is a pair of 1-D filters: the smoothing filter $p(k)$, and the gradient filter $d(k)$. Referring to equation (4) and assuming separability of the 3-D filters we get:

$$\{I * F_1\}(x, y, t)u^x(x, y, t) \quad +\{I * F_2\}(x, y, t)u^y(x, y, t) + \{I * F_3\}(x, y, t) = 0 \qquad (9)$$

$$\begin{aligned} \text{where:} \qquad & F_1(x, y, t) = d(x) * p(y) * p(t) \\ & F_2(x, y, t) = p(x) * d(y) * p(t) \\ & F_3(x, y, t) = p(x) * p(y) * d(t) \quad . \end{aligned}$$

Figure 5 plots the 3-D filters that are applied using the above described filters. This flow of operations is consistent with Equation 9.

Figure 6 shows the actual 1-D Simoncelli's filters for 9-taps: pre-smoother $p(x)$ and gradient $d(x)$. The filters are given in their continuous form, interpolated using the *sinc* function. Figure 7 shows the frequency response of the gradient filter, $D(\theta)$, compared to an analytic gradient of the smoother filter, $j\theta \cdot P(\theta)$. This time the error between these two is very small as compared to the result obtained with filters proposed by Barron et. al.

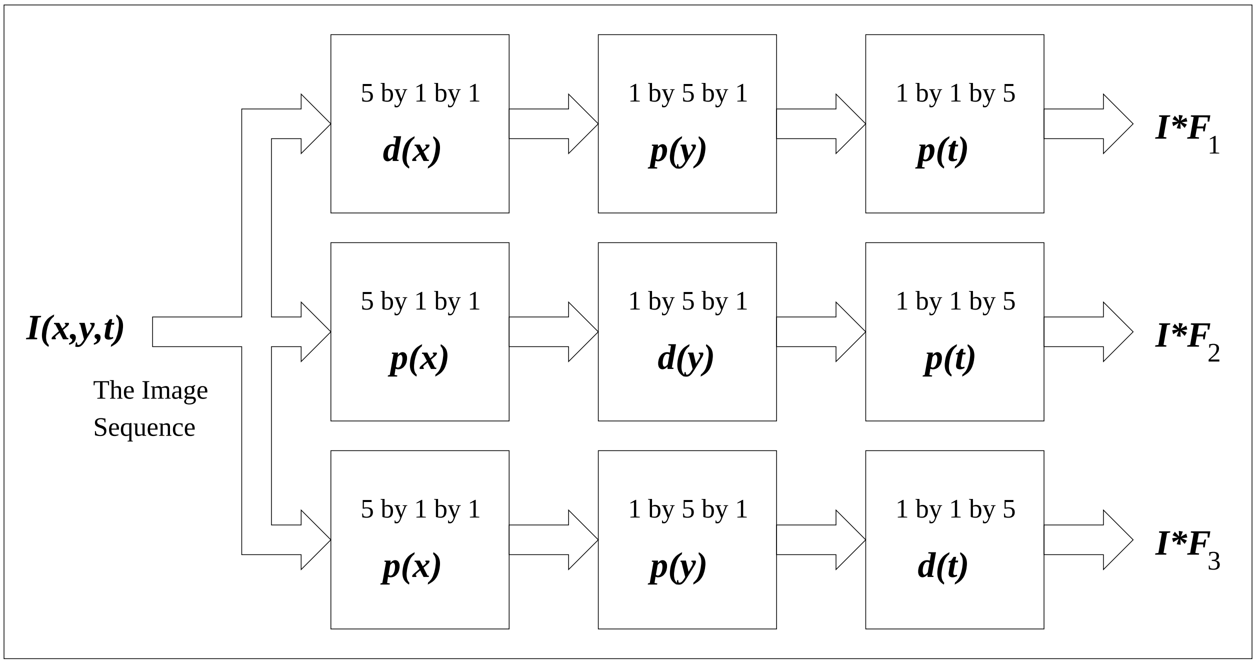
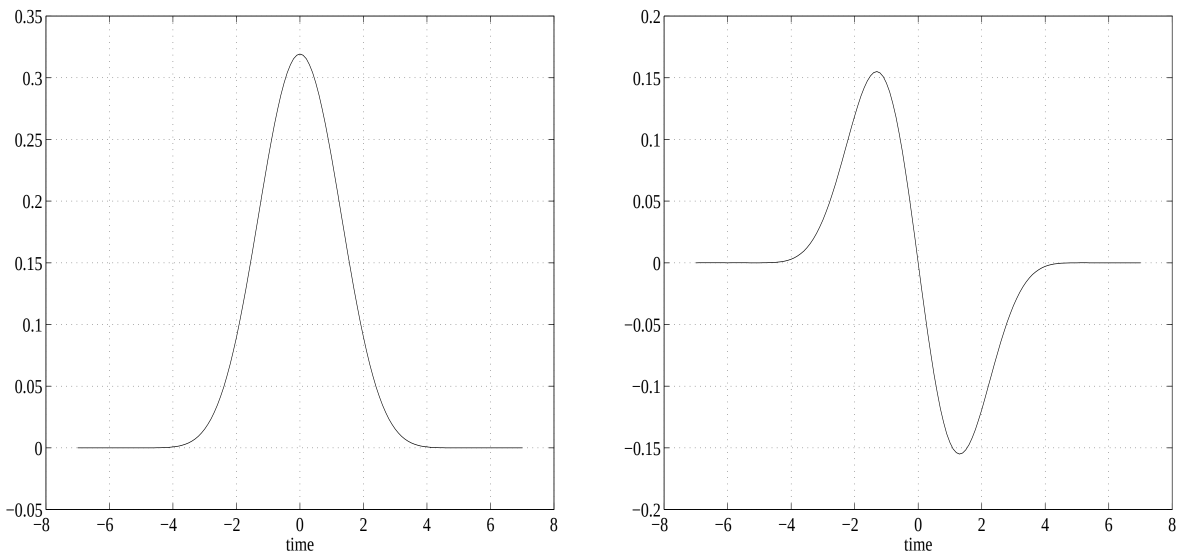


Figure 5: Simoncelli's proposed filters configuration



(a) (b)

Figure 6: Simoncelli's proposed filters (a) $p(x)$ and (b) $d(x)$

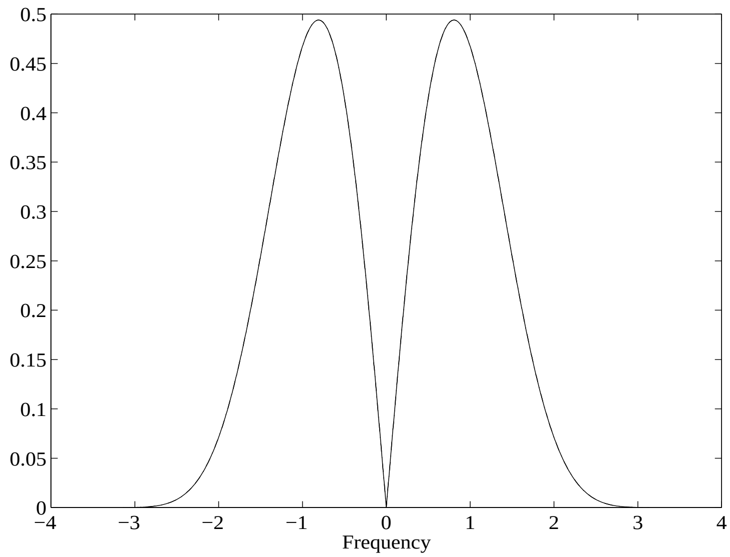


Figure 7: Simoncelli's proposed filters in the frequency domain $|D(\theta)|$ (solid line) and $|j\theta \cdot P(\theta)|$ (dashed line) (they are actually one on top of the other)

2.4 Existing Approaches - Summary

Existing methods for the design of filters to be used in optical flow estimation have been presented. The following concepts are beneficial and should be considered in any design procedure for derivative filters:

1. The filters should be FIR linear phase filters.
2. The design should relate to an interpolated version of the discrete image sequence in order to refer to the continuous grid.
3. The design of all the required filters should be done simultaneously.
4. The design procedure should consider the spectral construction of typical images, or better yet, the specific spectral structure of a given image sequence.

5. The design procedure should consider the number of taps allocated, and should exploit them in the best possible manner.

6. If possible, the design should yield separable filters which are easier to implement. Nevertheless, in the following discussion we shall deal with both separable and non-separable filters.

Beyond these specifications, there are a few more important requirements that should be met in order to obtain better results:

1. For optimal solution, the motion characteristics should be considered as well. For example, assume that apriori knowledge assures that the motion vector components are in the range of $[-2, 2]$. This knowledge should somehow influence the designed filters. No existing method considers this criterion.

2. Most important of all, minimization should be done with respect to the motion estimation error.

In the next section our new approach is presented. The suggested approach takes into consideration all the above requirements.

3 The Proposed Approach - The 1-Dimensional Case

3.1 Shiftable Filters

Assume that a pair of discrete 1-D images are given with small displacement between them. These images are denoted by $I_1(k)$ and $I_2(k) = I_1(k + \tau(k))$, $(k \in [1..L_I])$. Our goal is to estimate the space varying motion field $\tau(k)$. Using an interpolating function $b(x)$ we can recover the continuous images $\hat{I}_1(x)$ and $\hat{I}_2(x)$ by:

$$
\begin{aligned}
\hat{I}_1(x) &= \sum_{k=1}^{L_I} I_1(k) b(x-k) = b(x) * I_1(k) && (10)\\
\hat{I}_2(x) &= \sum_{k=1}^{L_I} I_2(k) b(x-k) = b(x) * I_2(k) = \hat{I}_1(x+\tau(x)) \quad .
\end{aligned}
$$

Assuming that $\tau(x)$ is small, and taking the first order Taylor expansion of the image $\hat{I}_2(x) = \hat{I}_1(x+\tau(x))$ around x yields:

$$
\begin{aligned}
\hat{I}_2(x) &= \hat{I}_1(x+\tau(x)) = \hat{I}_1(x) + \frac{\partial \hat{I}_1(x)}{\partial x}\tau(x) + o\{\tau(x)^2\} && (11)\\
\Rightarrow b(x) * I_2(k) &= b(x) * I_1(k) + \tau(x) \cdot \frac{\partial b(x)}{\partial x} * I_1(k) + o\{\tau(x)^2\} \quad .
\end{aligned}
$$

In the above expansion, we have neglected the terms which correspond to the derivative of $\tau(x)$ since $\tau(x)$ is assumed to be locally smooth.

This equation is actually the BCCE as introduced in the previous section. It contains an error term which is quadratic with respect to the motion vector $\tau(x)$, resulting from the convergence property of the Taylor series. This error term means that small displacements are estimated more accurately than larger ones. In general, this property is too restrictive. There should be no reason to assume that small displacements need to be more accurately estimated than larger ones.

The above equation involves three filters; one filter is applied to I_2 (the filter $b(x)$), and two filters to I_1 (the pair $b(x)$ and $\partial b(x)/\partial x$). Two of these three filters are identical, and the third is a pure derivative of the formers. Again, this choice of filters is too restrictive. In our approach, three (might be different) filters are designed to estimate the optical flow in an optimal manner. In this general form, we are looking for three filter $\hat{m}(x)$, $\hat{h}(x)$, and $\hat{g}(x)$, such that:

$$\hat{m}(x) * I_2(k) = \hat{h}(x) * I_1(k) + \tau(x) \cdot \hat{g}(x) * I_1(k) + \epsilon(x, \tau(x)) \quad . \tag{12}$$

Here, $\hat{m}(x)$, $\hat{h}(x)$, and $\hat{g}(x)$ are represented as continuous filters. However, note that these filters are actually discrete FIR kernels that have been interpolated to form continuous filters:

$$\begin{aligned} \hat{m}(x) &= \sum_{k=-L}^{L} m(k) b(x-k) = \mathbf{m}^T \mathbf{b}(\mathbf{x}) \quad , \\ \hat{h}(x) &= \sum_{k=-L}^{L} h(k) b(x-k) = \mathbf{h}^T \mathbf{b}(\mathbf{x}) \quad , \\ \hat{g}(x) &= \sum_{k=-L}^{L} g(k) b(x-k) = \mathbf{g}^T \mathbf{b}(\mathbf{x}) \quad , \end{aligned} \tag{13}$$

where

$$\begin{aligned} \mathbf{m}^T &= [m(-L), ..., m(L)] \quad , \\ \mathbf{h}^T &= [h(-L), ..., h(L)] \quad , \\ \mathbf{g}^T &= [g(-L), ..., g(L)] \quad , \\ \mathbf{b}(\mathbf{x})^T &= [b(x+L), ..., b(x-L)] \quad . \end{aligned}$$

Although the number of taps for these filters is $2L+1$, in the general case, we can have an even number of taps as well.

In order to obtain the optimal triplet $\hat{m}(x)$, $\hat{h}(x)$, and $\hat{g}(x)$, the above equation should hold approximately for each position x and for any possible motion field $\tau(x)$. In other words, the following error term should be minimized:

$$\Gamma(\mathbf{m}, \mathbf{h}, \mathbf{g}) = E_{\tau(x)} \left\{ \int_x \epsilon(x, \tau(x))^2 dx \right\} \tag{14}$$

where the term $\epsilon(x, \tau(x))$ is as defined in Equation 12, and $E_z\{f(z)\}$ is the expected value of $f(z)$. While the minimization over all x is simple, the minimization over all possible $\tau(x)$ is problematic due to the necessity to specify an infinite number of possible motion patterns. To avoid this problem, the motion field can be expressed as a parametric function where the minimization is done over some admissible range of the parameters. A simple parametric expansion using the Taylor series gives:

$$\tau(x) = \tau(x_0) + \tau_x(x_0)x + o(x^2) \quad . \tag{15}$$

The zero-th order approximation approximates $\tau(x)$ as a pure translation: $\tau(x) \approx \tau(x_0) \equiv \tau$. The first order approximation adds an additional term $\tau_x(x_0)x$ resembling a scaling factor to the motion field. At this point, we will concentrate on the pure translation model. Note that this model assumes a translated motion only *locally* (in the range of the filter support). The global motion doesn't have to be a pure translation. However, this approximation is valid only for smooth motion fileds. More complicated motion model can be treated similarly, and will be dealt with in a later publication.

According to the above assumption, we get that $\hat{I}_2(x) = \hat{I}_1(x + \tau(x)) \approx \hat{I}_1(x + \tau)$. That in turn means that applying the shift invariant filter $\hat{m}(x)$ to the image $I_2(k)$ (which is equal to $\hat{I}_1(x + \tau)$), can be performed equivalently by first applying the (shift invariant) transformation to the filter, and then convolving with the original image: $\hat{m}(x) * I_2(k) = \hat{m}(x) * I_1(k + \tau) = \hat{m}(x + \tau) * I_1(k)$. Therefore, we can rewrite Equation 12 with respect to I_1 solely:

$$\begin{aligned} \hat{m}(x+\tau) * I_1(k) &= \hat{h}(x) * I_1(k) + \tau \cdot \hat{g}(x) * I_1(k) + \epsilon(x, \tau) = \\ &= [\hat{h}(x) + \tau \cdot \hat{g}(x)] * I_1(k) + \epsilon(x, \tau) \quad . \end{aligned} \tag{16}$$

In the above equation, the function $\hat{m}(x)$, shifted by τ, is approximated by $\hat{h}(x)+\hat{g}(x)\tau$. This resembles the concept of **"Shiftable Filters"** as defined in [11, 12], where a transformed version of a function is expressed as a linear sum of a set of basis functions.

Assuming that τ is bounded within the range $|\tau| \leq D$, and that the interpolating function is $b(x) = sinc(x) = sin(\pi x)/(\pi x)$, the error term $\Gamma(\mathbf{m}, \mathbf{h}, \mathbf{g})$ can be rewritten as:

$$\begin{aligned} \Gamma(\mathbf{m}, \mathbf{h}, \mathbf{g}) &= \int_x \int_{\tau=-D}^{D} \epsilon^2(x,\tau) dx d\tau = \\ &= \int_x \int_{\tau=-D}^{D} \left[I_1(k) * \left(\hat{m}(x+\tau) - \hat{h}(x) - \hat{g}(x)\tau \right) \right]^2 dx d\tau \\ &= \sum_{k=-\infty}^{\infty} \int_{\tau=-D}^{D} \left[I_1(k) * (m(k+\tau) - h(k) - g(k)\tau) \right]^2 d\tau \end{aligned} \tag{17}$$

where $\hat{m}(x), \hat{h}(x), \hat{g}(x)$ are as defined in Equation 13. The fact that the integration over all x is equivalent to the summation over all integer values of x is based on the choice of the interpolating function. A proof of this property is given in Appendix A.

Note that if we have a prior knowledge about the distribution of the motion vectors (for example, knowing that smaller values are more probable), we can add a weighting function, $w(\tau)$, into the internal integration, requiring smaller errors for the more probable motion vectors.

Using Parseval's theorem, the design goal can be re-formulated in the frequency domain:

$$\begin{aligned} \Gamma(\mathbf{m}, \mathbf{h}, \mathbf{g}) &= \int_{\theta=-\pi}^{\pi} \int_{\tau=-D}^{D} \Im_\theta \left[\epsilon(k,\tau) \right]^2 d\theta d\tau = \\ &= \int_{\theta=-\pi}^{\pi} \int_{\tau=-D}^{D} |I_1(\theta)|^2 \left| \left(e^{j\theta\tau} M(\theta) - H(\theta) - G(\theta)\tau \right) \right|^2 d\theta d\tau \\ &= \int_{\theta=-\pi}^{\pi} \int_{\tau=-D}^{D} |I_1(\theta)|^2 \left| \Upsilon(\theta,\tau) \right|^2 d\theta d\tau \end{aligned} \tag{18}$$

where $\Im_\theta[f(k)]$ stands for the Discrete Fourier Transform (DFT) of $f(k)$. The terms $M(\theta)$, $H(\theta)$, $G(\theta)$, and $I_1(\theta)$ are the DFT of $m(k)$, $h(k)$, $g(k)$, and $I_1(k)$, respectively. $I_1(\theta)$ is normalized to be in the range $0 \leq |I_1(\theta)| \leq 1$.

The error term $\Gamma(\mathbf{m}, \mathbf{h}, \mathbf{g})$ is constructed as a Weighted Least Squares (WLS) of the error $\Upsilon(\theta, \tau)$ summed over all the frequencies θ and all admissible τ, and weighted by the spectra of the image $I_1(\theta)$. This term exhibits the desired property that smaller error $|\Upsilon(\theta, \tau)|$ is required in frequencies where $|I_1(\theta)|$ is higher. However, this property can also be acheived by attenuating the energy of the filters $M(\theta)$, $H(\theta)$ and $G(\theta)$ in frequency ranges where $|I_1(\theta)|$ has low energy. This in turn means that applying these filters to the image give a non-informative result. To correct this behavior we add an additional term that requires the energy of the filters to be at frequencies where $|I_1(\theta)|$ is high. The overall penalty function is therefore:

$$\begin{aligned}\Gamma_{\text{total}}(\mathbf{m}, \mathbf{h}, \mathbf{g}) &= \int_{\theta=-\pi}^{\pi} \int_{\tau=-D}^{D} |I_1(\theta)|^2 \left|\Upsilon(\theta, \tau)\right|^2 d\theta d\tau + \\ &\quad + \alpha \int_{\theta=-\pi}^{\pi} \int_{\tau=-D}^{D} \left[1 - |I_1(\theta)|^2\right] |\Upsilon(\theta, \tau)|^2 d\theta d\tau = \\ &= \int_{\theta=-\pi}^{\pi} \int_{\tau=-D}^{D} \left[\alpha + (1-\alpha)|I_1(\theta)|^2\right] |\Upsilon(\theta, \tau)|^2 d\theta d\tau \quad .\end{aligned} \tag{19}$$

The parameter α controls the relative weight between $\Gamma(\mathbf{m}, \mathbf{h}, \mathbf{g})$ and the new penalty, and should be in the range $0 \leq \alpha \leq \infty$. For $\alpha = 0$, we get the previous error term; for $\alpha = 1$, the error term contains a flat frequency weight.

The proposed criterion uses knowledge of the maximal length of the motion vectors, and possibly, their distribution. Additionally, properties of the input image are considered in the minimization procedure as well. This is a natural and beneficial property since we expect that different images would require different filters. From an implementation point of view, however, for a given pair of images, we have to compute their corresponding optimal filters, and then estimate their motion field. To avoid the computations of the specific filters for any input images we can design a set of filters which should match the spectral characteristics of typical natural images (such as $1/\theta$ as proposed by Simoncelli [8]).

The minimization term in Equation 19 can be rewritten in a bilinear form

$$\Gamma_{\text{total}}(\mathbf{m}, \mathbf{h}, \mathbf{g}) = \mathbf{x}^H \mathbf{R} \mathbf{x} \quad ,$$

where the vector $\mathbf{x}$ contains all the filters coefficients, and the operation $\mathbf{x}^H$ is the complex conjugate of $\mathbf{x}$:

$$\mathbf{x}^H = [\mathbf{m}^H, \mathbf{h}^H, \mathbf{g}^H] \quad . \tag{20}$$

The matrix $\mathbf{R}$ in a $(3L+3) \times (3L+3)$ positive definite matrix, which depends on the interpolating function $b(x)$, the maximal motion D, and the spectral characteristics of the image $I_1(k)$. Appendix B describes the content of this matrix for two cases; where the original three filters $\mathbf{m}$,$\mathbf{h}$ and $\mathbf{g}$ are different from each other, and where $\mathbf{m} = \mathbf{h}$.

In order to avoid the trivial solution, the minimization is performed with the following constraint:

$$\text{Minimize:}\, \mathbf{x}^H \mathbf{R} \mathbf{x} \quad \textbf{s.t. :} \quad \mathbf{x^H x = 1} \quad . \tag{21}$$

The solution of the above problem is the eigenvector of the matrix $\mathbf{R}$ corresponding to the smallest eigenvalue, and can be obtained using the SVD decomposition [13].

3.2 The Relationship to Simoncelli's Filters

As mentioned in Section 2.3, Simoncelli also proposed a frequency domain WLS criteria for the design of optimal filters. This subsection explores the connection between his approach and ours. It can be shown that the filters suggested by Simoncelli are actually a special case of those designed by our approach where:

1. $\mathbf{m} = \mathbf{h}$,
2. $b(x) = sinc(x) = \frac{sin(\pi x)}{\pi x}$,
3. $|I_1(\theta)|^2 = \frac{1}{\sqrt{|\theta|}}$,
4. $D \rightarrow 0$, and
5. $\alpha = 0$.

This can be shown using the truncated Taylor series $e^{j\theta\tau} \approx 1 + j\theta\tau$, which is a good approximation for small τ ($\tau \in [-D, D]$). Using this relation and the above assumptions within our error term (Equation. 19) we get:

$$\begin{aligned}
\Gamma_{\text{total}}(\mathbf{m}, \mathbf{h}, \mathbf{g}) &= \int_{\theta=-\pi}^{\pi} \int_{\tau=-D}^{D} \frac{1}{\sqrt{|\theta|}} \left|(1 + j\theta\tau - 1)M(\theta) - G(\theta)\tau\right|^2 d\theta d\tau = \\
&= \int_{\theta=-\pi}^{\pi} \frac{1}{\sqrt{|\theta|}} \left|j\theta M(\theta) - G(\theta)\right|^2 \int_{\tau=-D}^{D} \tau^2 d\tau d\theta = \\
&= \frac{2D^3}{3} \int_{\theta=-\pi}^{\pi} \frac{1}{\sqrt{|\theta|}} \left|j\theta M(\theta) - G(\theta)\right|^2 d\theta \quad .
\end{aligned} \tag{22}$$

Up to a constant, we got the same optimality criterion as in Equation 8. Since Simoncelli filters correspond to infitesimal motion, the estimation of larger motion vectors are subject to larger errors. Furthermore, since Simoncelli's method corresponds to $\alpha = 0$, in some cases (large number of taps or specific band limited images) his filters give poor motion estimation performance.

An interesting question in this context is the following: Assume that it is known that $D \to 0$ (very small motion vectors, as assumed by Simoncelli), $\alpha = 1$ (which means that the weights are 1 for all frequencies, which is better suited for filters design), and that $b(x) = sinc(x) = \frac{sin(\pi x)}{\pi x}$ (the trivial choice). The question is: What are the optimal filters **m**, **h** and **g** in this case? Appendix C treats this question, and shows that these assumptions lead to the following result:

1. $\mathbf{m} = \mathbf{h}$,

2. The filter **g** is given by the sampled derivative of the interpolated filter **m**:

$$g(j) = \sum_{k=-L}^{L} m(k) \frac{(-1)^{j-k}}{j-k} \qquad -L \le j \le L \tag{23}$$

   where $(-1)^k/k$ is the sampled derivative of the Sinc interpolation function (see Appendix C).

3. The filter **m** can be approximated by the raised cosine function:

$$m(k) \approx 1 + \cos\left[\frac{k\pi}{L+1}\right] \qquad -L \le k \le L \ . \tag{24}$$

The implication of the above result is that optimal filters according to the Simoncelli's definition (changing only the frequency weight, which, as will be shown later, is better) can be obtained analytically in a very simple manner, overcoming the need for a numerical filters design procedure.

4 Examples

In this section we present several examples which demonstrate the ability of the proposed optimal filters to give better motion estimation performance. Figure 8 shows the three filters (9-taps) $\mathbf{m}$, $\mathbf{h}$ and $\mathbf{g}$ obtained for 3 different values of D (the maximum expected motion vector): $D = 0.1$, 2 and 4 pixels. In all these cases, the filters where obtained with $\alpha = 1$ (which means that no frequency weight is involved). For comparison, Simoncelli's 9-taps filters ($\mathbf{m} = \mathbf{h}$) are given as well (with $\alpha = 0$). All these graphs plot interolated versions of the discrete filters (using $Sinc$ interpolations).

As expected, when D is very small we get $\mathbf{m} = \mathbf{h}$. Another property that can be seen from these graphs is the better exploitation of the filters support. As D gets bigger, the obtained filters (with the same number of taps) are getting wider (for the same number of taps).

Recall that the proposed filters are the ones which minimize the error
$(m(x+\tau) - h(x) - g(x)\tau)^2$, averaged over all x and all τ in the range $[-D, D]$. Figure 9 shows a graph of this error as a function of τ for three sets of filters:

1. Barron's 9-taps filters.
2. Simoncelli's 9-taps filters.
3. The optimal 9-taps filters for $D = 2$ and $\alpha = 1$.

As can be seen, Simoncelli's and Barron's filters have zero error for $\tau = 0$, whereas the optimal filters give non-zero error. This comes from the fact that in the optimal case $\mathbf{m} \neq \mathbf{h}$. Beyond that, notice that the overall error is much smaller with the optimal filters (because

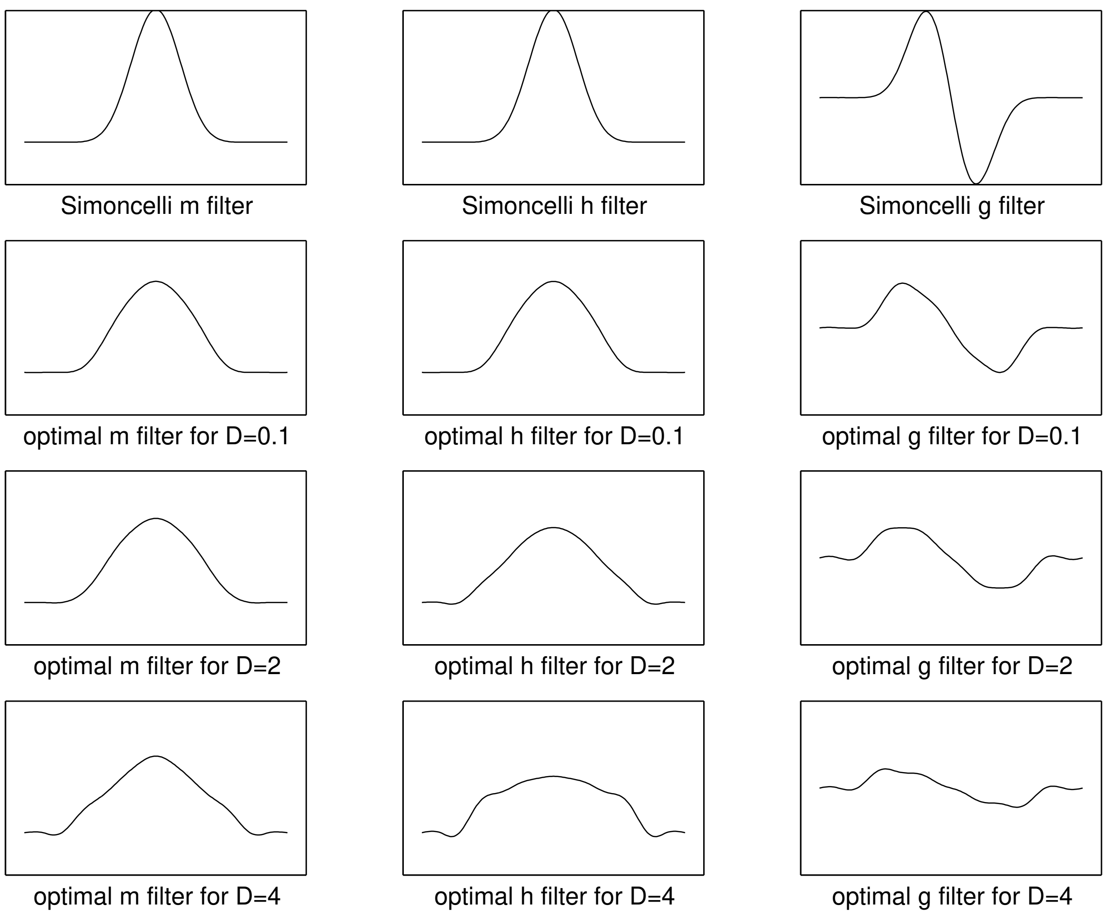


Figure 8: Simoncelli's filters (top row) and the optimal filters for $D = 0.1$ (second row), 2 (third row) and 4 pixels (last raw).

of moderate errors for large values of τ). The average errors (the value of the integral) for Barron, Simoncelli and the optimal filters are 0.172, 0.1782 and 0.0961 respectively. This also suggests that Simoncelli's filters are comparable to Barron filters.

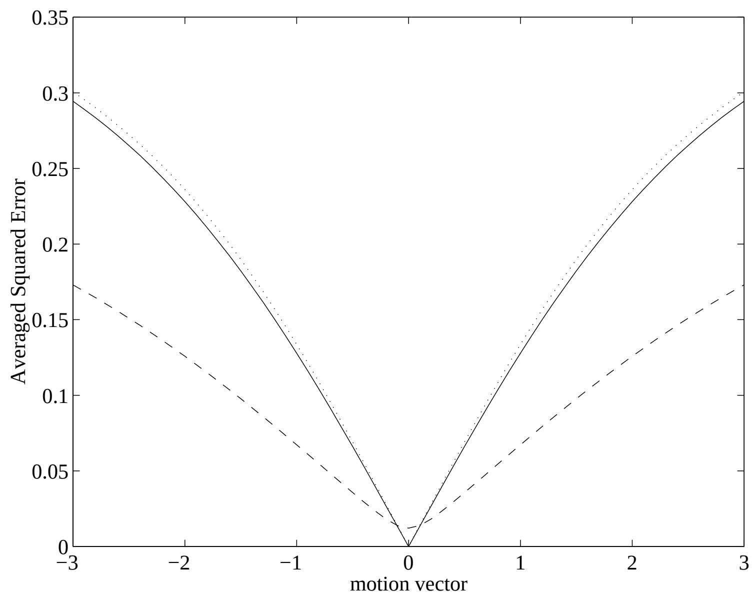


Figure 9: The error $\left[\int_x \left(m(x+\tau) - h(x) - g(x)\tau\right)^2 dx\right]^{0.5}$ as a function of τ for Barron (solid line), Simoncelli's (dotted line) and the optimal filters (dashed line).

It should be noted that the above comparison does not prove that the obtained filters are indeed the optimal ones, since optimality (or at least better performance) should be demonstrated with respect to optical flow estimation errors. This is the reasoning behind the following several examples. We took random pairs of vectors of length 160 having relative global translation between them in the range $\tau \in [-3, 3]$. We estimated the motion field using two different techniques: Lucas and Kanade's [14], and Horn and Schunck's [4] algorithms. We implemented these two algorithms with three different sets of filters: Barron's filters, Simoncelli's filters and our optimal filters.

Lucas and Kanade's algorithm suggests that for each pixel, the BCCE of its local neighborhood should be combined to form a least-squares solution. We have used uniform weights over a 5-pixel neighborhood [14, 3]. Horn and Schunck's algorithm reconstructs the motion vectors by adding a smoothness penalty. We have used the Laplacian operator for this penalty with $\beta = 300$. In order to reduce randomness effects, 10 such pairs were generated per each motion vector value. Figure 10 shows one of such ($\tau = 2$ pixels). Figure 11 shows the averaged absolute motion estimated error with Horn and Schunck's, and Lucas and Kanade's algorithms. Errors are presented for the Barron filters, Simoncelli's filters and the optimal filters.

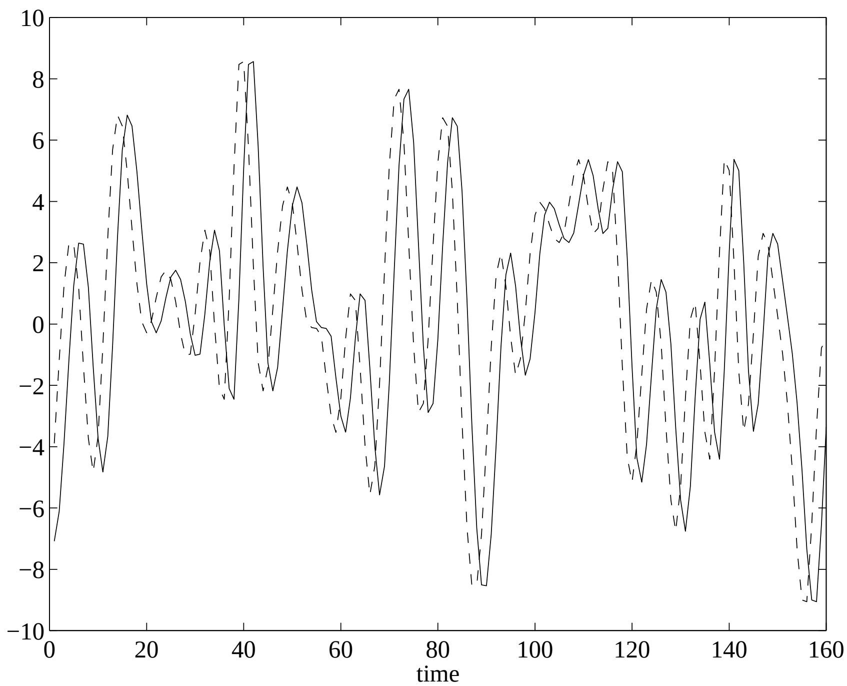
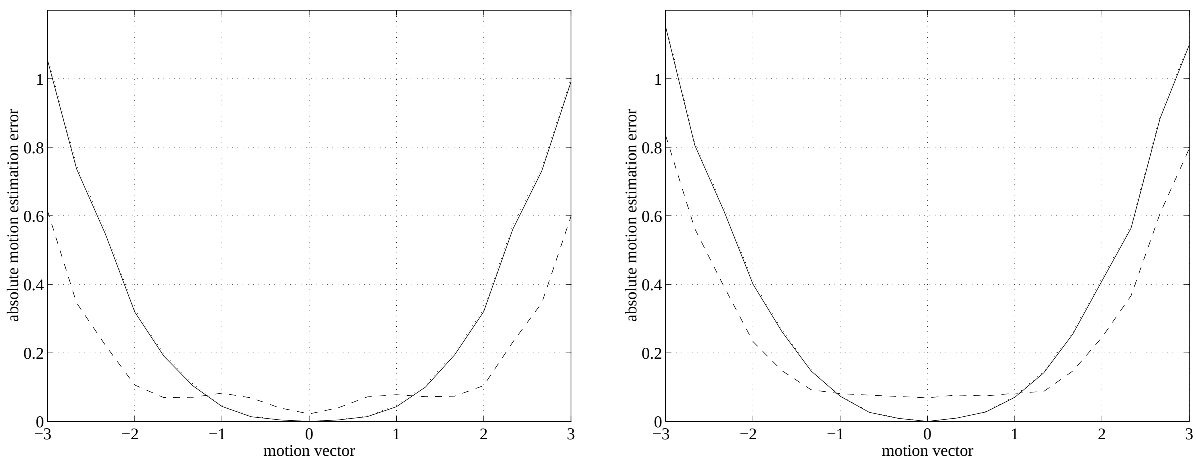


Figure 10: An example of a pair of vectors used for the demonstrated tests.

The results show that the graph in Figure 9 indeed corroborates the actual errors obtained in the motion estimation results. Figure 12 shows the accumulative error as a function of D



(a) **(b)**

Figure 11: The average absolute motion estimation error for Barron's, Simoncelli's and the optimal filters - (a) Horn and Schunck algorithm, (b) Lucas and Kanade algorithm. Barron (solid line), Simoncelli's (dotted line) and optimal filters (dashed line).

in the range $D \in [0, 3]$, based on the results shown in Figure 11 (this graph shows the area beneath the error function in the range $\tau \in [-D, D]$ for different values of D).

Another issue which was raised in the previous sub-section is the choice of the parameter α. We have claimed that using $\alpha = 0$ (as in Simoncelli's case) sometimes (depending on the given image, number of taps, and the maximal motion value) the results are non-informative filters. The following Figures describes the influence of α on the optimal filters. Figure 13 shows 3 sets of 9-taps optimal filters $\mathbf{m}$, $\mathbf{h}$ and $\mathbf{g}$ for $D = 2$ and $\alpha = 0$, 0.01, 1, and 10.

These graphs revels that for $\alpha = 0$ the obtained filters are modulated, in order to be located in frequencies where the image spectrum is low. When α is increased (even for very small values), this problem is overcomed. Note that the obtained filters are robust with respect to the choice of α (for non-zero values of α). This of-course implies that $\alpha = 1$ is a very

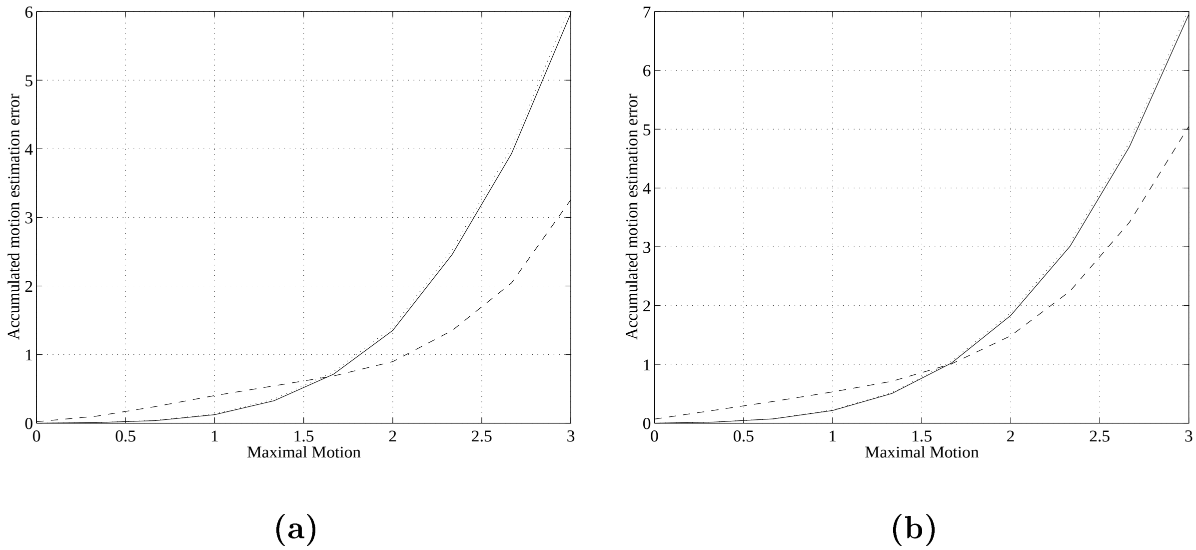


Figure 12: The accumulated error of the motion estimation for Barron's, Simoncelli's and the optimal filters - (a) Horn and Schunck algorithm, (b) Lucas and Kanade algorithm. Barron (solid line), Simoncelli's (dotted line) and optimal filters (dashed line).

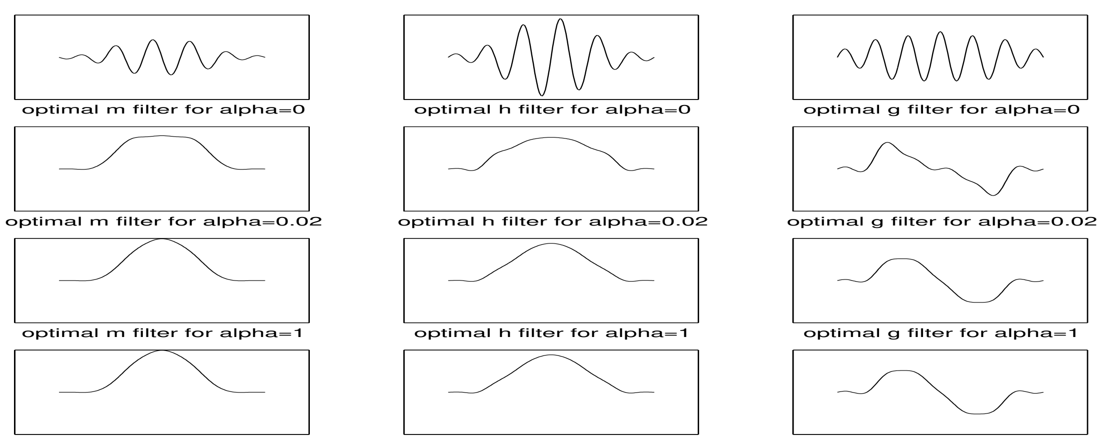
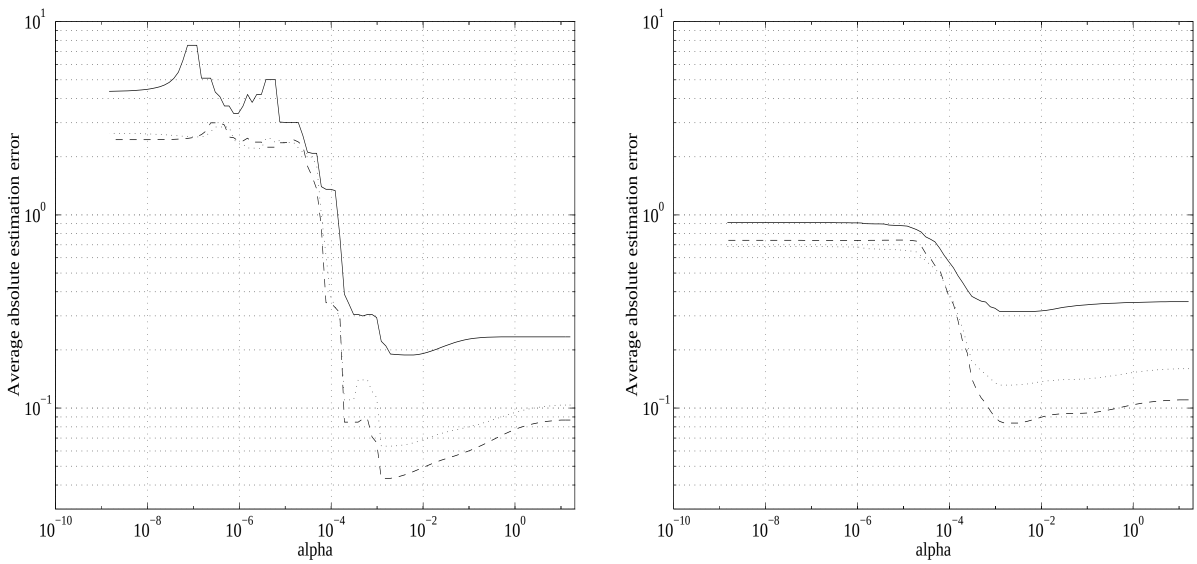


Figure 13: 9-taps optimal filters **m**, **h** and **g** for $D = 2$ and $\alpha = 0$ (first row), 0.02 (second row), 1 (third row), and 10 (last row).

good choice - it results in uniform frequency weight which no longer depends on the image spectrum. This in turn means that this spectrum need not be evaluated. Beyond that, we see that the obtained filters are general and serve all the images, as they do not depend on the image characteristics.

Figure 14 displays the average absolute motion estimation error as a function of α for Horn and Schunck's and Lucas and Kanade's algorithms. These graphs were obtained using optimal filters designed to $D = 2$. Again, each point in these graphs was obtained by averaging on 10 different random vector pairs.



(a) (b)

Figure 14: The average absolute motion estimation error as a function of α for three different integration ranges: Solid line - $\tau \in [-3, 3]$, Dotted line - $\tau \in [-2, 2]$, Dashed line - $\tau \in [-1, 1]$. (a) Horn and Schunck algorithm, (b) Lucas and Kanade algorithm.

We can conclude from these two graphs two things: (i) Very low values of α gives very low quality estimate results, and (ii) There is indeed robustness with respect to the value of α

beyond some very low threshold.

As a final stage of this section, we describe the performance of the proposed filters on true images. In this context we first have to define how the 2-D (in the case where two images are given) and 3-D (in a case where an image sequence is given) filters are to be composed based on the obtained 1-D kernels. Thre 2-D case can be modeled by the equation: $I_2(k,j) = I_1(k+\tau_x(k,j), j+\tau_y(k,j))$. Similar to Equation 12, we have the following general BCCE relationship:

$$\begin{aligned} \hat{m}_{2D}(x,y) * I_2(k,j) \;\; &\approx \;\; \hat{h}_{2D}(x,y) * I_1(k,j) + \\ &+\tau_x(x,y)\cdot\hat{g}_{2D}(x,y) * I_1(k,j) + \tau_y(x,y)\cdot\hat{f}_{2D}(x,y) * I_1(k,j) \end{aligned} \tag{25}$$

Notice that now there are four filters involved, since the motion vector has two components - $[\tau_x(x,y), \tau_y(x,y)]$. These four filters are constructed from the 1-D filters $\hat{m}(x)$, $\hat{h}(x)$ and $\hat{g}(x)$ by the following relashionships:

$$\begin{aligned} \hat{m}_{2D}(x,y) &= \hat{m}(x)\cdot\hat{m}(y) \\ \hat{h}_{2D}(x,y) &= \hat{h}(x)\cdot\hat{h}(y) \\ \hat{g}_{2D}(x,y) &= \hat{g}(x)\cdot\hat{h}(y) \\ \hat{f}_{2D}(x,y) &= \hat{h}(x)\cdot\hat{g}(y) \end{aligned} \tag{26}$$

Similarly, the 3-D BCCE and the filters which corresponds to a volume of images is given by:

$$\begin{aligned} \hat{m}_{3D}(x,y,t) * I(k,j,n+1) \;\; &\approx \;\; \hat{h}_{3D}(x,y) * I(k,j,n) + \\ &+\tau_x(x,y,t)\cdot\hat{g}_{3D}(x,y,t) * I(k,j,n) + \\ &+\tau_y(x,y,t)\cdot\hat{f}_{3D}(x,y,t) * I(k,j,n) \end{aligned} \tag{27}$$

where

$$\begin{aligned}
\hat{m}_{3D}(x,y,t) &= \hat{m}(x)\cdot\hat{m}(y)\cdot\hat{m}(t) \\
\hat{h}_{3D}(x,y,t) &= \hat{h}(x)\cdot\hat{h}(y)\cdot\hat{m}(t) \\
\hat{g}_{3D}(x,y,t) &= \hat{g}(x)\cdot\hat{h}(y)\cdot\hat{m}(t) \\
\hat{f}_{3D}(x,y,t) &= \hat{h}(x)\cdot\hat{g}(y)\cdot\hat{m}(t)
\end{aligned} \tag{28}$$

Applying $\hat{m}(t)$ in the temporal axis plays the roll of the pre-smoothing operator which is typically used. Figure 15 shows this composition of the filters.

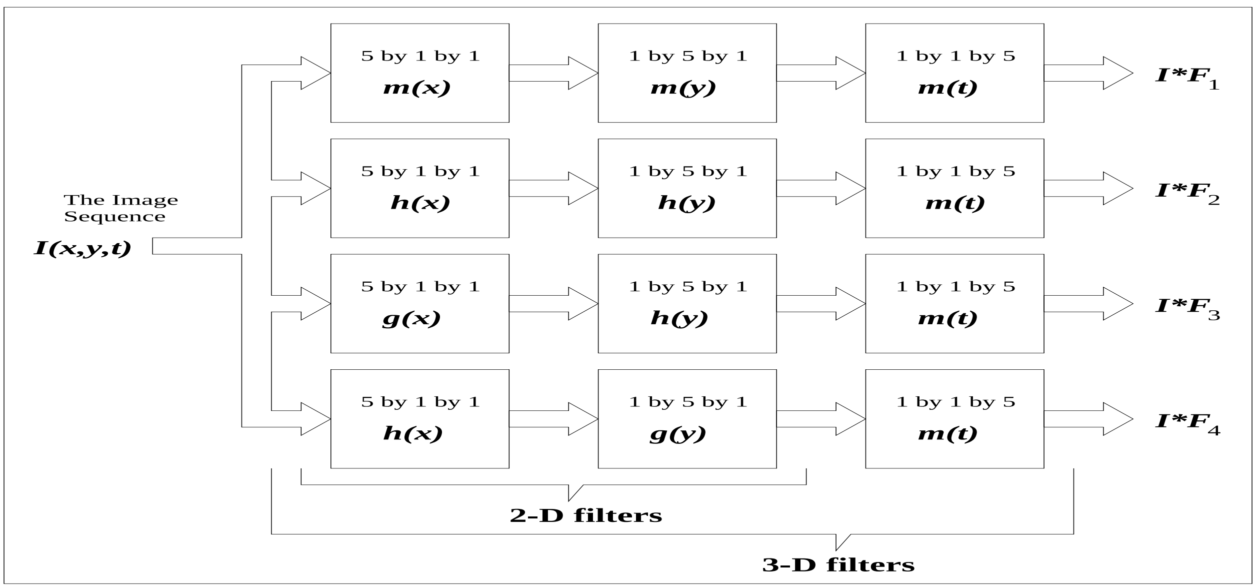


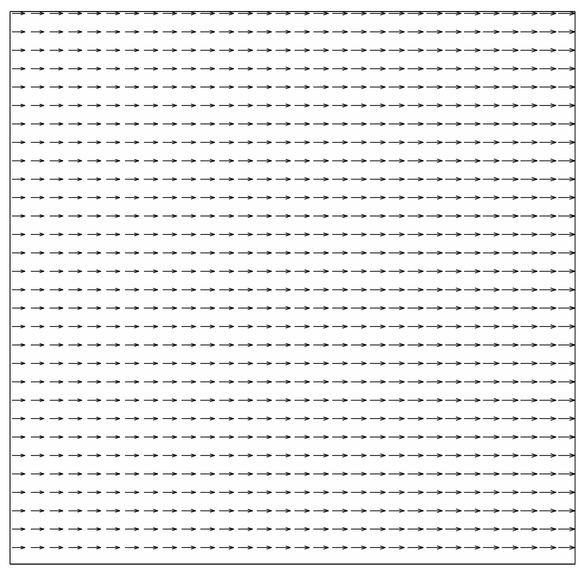
Figure 15: The proposed optimal filters configuration for the 2-D and 3-D cases.

The Barron's filters were taken to be 11-taps pre-smoothing and 5 taps gradient filter. Thus, the filter sizes are $\hat{m}_{2D} - [11\times 11]$, $\hat{h}_{2D} - [11\times 11]$, $\hat{g}_{2D} - [15\times 11]$, and $\hat{f}_{2D} - [11\times 15]$. In order to apply an objective comparison, we have used $[11\times 11]$ taps optimal filters. These filters were designed as 1-D 11-taps filters with $\alpha = 1$ and $D = 2$.

We have estimated the motion between three different pairs of images. We have not simulated the performance of the Simoncelli's filters since for this number of taps we got that the obtained Simoncelli's filters are non-symmetric, exhibiting instability of the solution. Beyond that, based on the 1-D results we expect the performance of Simoncelli's filters to be similar to Barron's.

In all cases we have constructed 2-D filters from the given 1-D ones, based on the above composition. Figures 16, 17 and 18 show the three images we have used, together with the true optical flow. These sequences, together with their true flow, were taken from Barron's WEB-site, and are called Translating Tree, Diverging Tree and Yosemite, respectively.

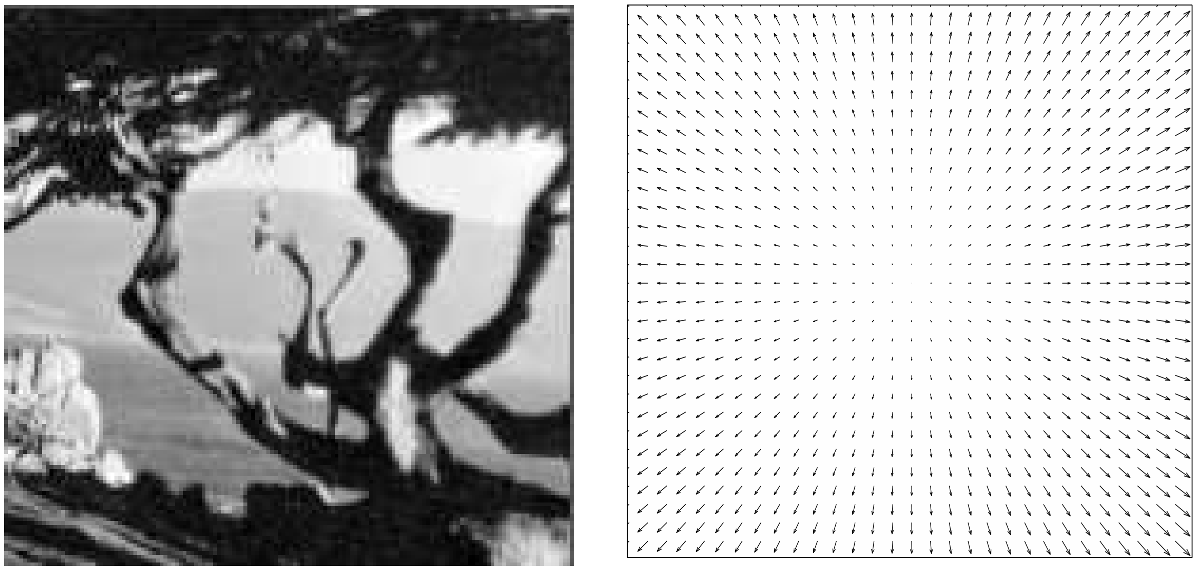
(a)



(b)

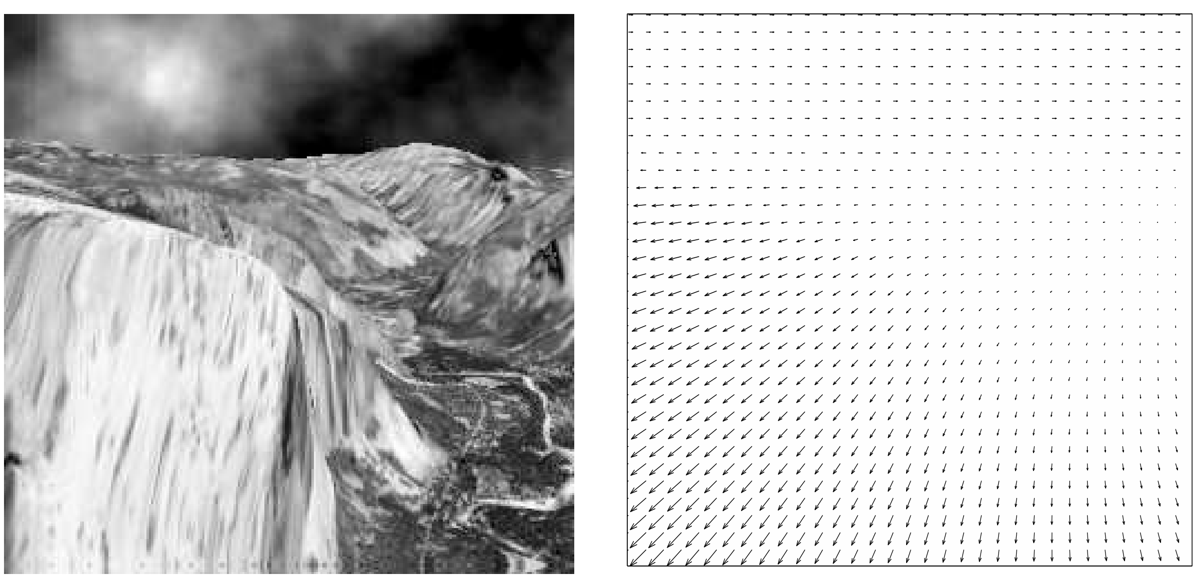
Figure 16: The first test sequence (Translating Tree) with its true optical flow.

In our simulations we have estimated the motion using Lucas and Kanade's [14] algorithm with neighborhood of $[7 \times 7]$ pixels, weighted uniformly (adequate for smooth motion flow). Figure 19 the estimated results for the first sequence. The two left maps correspond to full density optical flow, and the two right maps correspond to half density optical flow (obtained



(a) **(b)**

Figure 17: The second test sequence (Diverging Tree) with its true optical flow.



(a) **(b)**

Figure 18: The third test sequence (Yosemite) with its true optical flow.

by removing half of the vectors which were assigned low confidence measure). The upper row maps corrspond to Barron's filters, and the lower to the suggested optimal filters. Similar results are shown for the second and the third sequences in Figures 20 and 21.

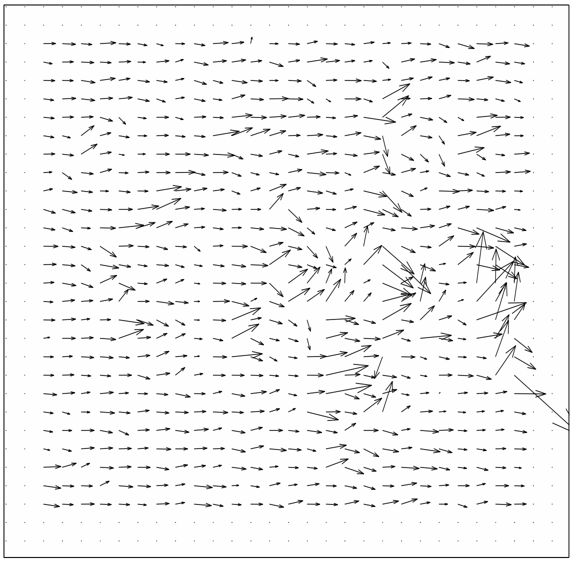
Tables 1,2 and 3 summarizes the obtained results for the three sequences. Per each sequence we have computed the average angular error [3], together with its variance, for varying density values. We have also supplied the Mean Squared Error between the true and the estimated flow for varying density values.

	Barron's Filters			Optimal Filters		
Optical Flow Density	Average Angular Error [degrees]	STD of Angular Error [degrees]	Mean Squared Error [pixels]	Average Angular Error [degrees]	STD of Angular Error [degrees]	Mean Squared Error [pixels]
0.15	12.0354	8.0263	0.6462	9.8192	8.7391	0.6169
0.30	13.9237	10.6659	0.8096	10.7882	10.2932	0.7601
0.45	15.4517	12.9075	1.0059	11.7653	11.9096	0.9069
0.60	16.0816	14.1794	1.1644	12.5050	13.2483	1.0875
0.75	16.1933	14.6669	1.2647	12.8682	14.0673	1.1823

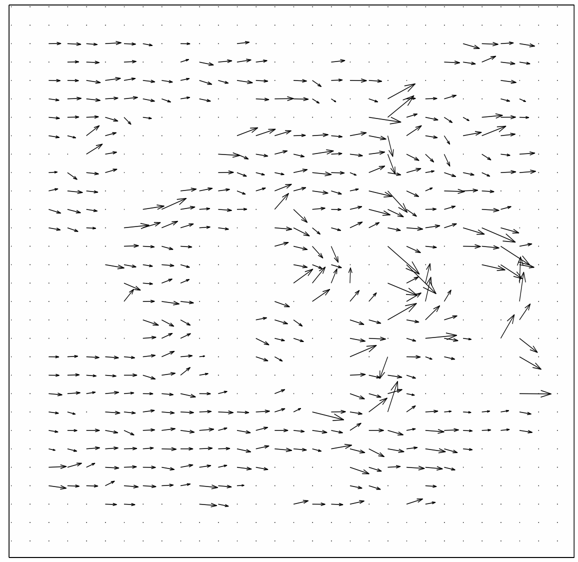
Table 1: Motion estimation results for the first sequence - Translating Tree

	Barron's Filters			Optimal Filters		
Optical Flow Density	Average Angular Error [degrees]	STD of Angular Error [degrees]	Mean Squared Error [pixels]	Average Angular Error [degrees]	STD of Angular Error [degrees]	Mean Squared Error [pixels]
0.15	4.9651	3.1782	0.1452	4.3478	2.6481	0.1298
0.30	5.9045	4.4907	0.1876	5.0700	3.6257	0.1699
0.45	7.0302	6.1049	0.2480	5.6658	4.4400	0.2011
0.60	8.0989	7.1865	0.3009	6.4420	5.5792	0.2380
0.75	8.7863	7.4659	0.3185	7.6778	7.3792	0.2971

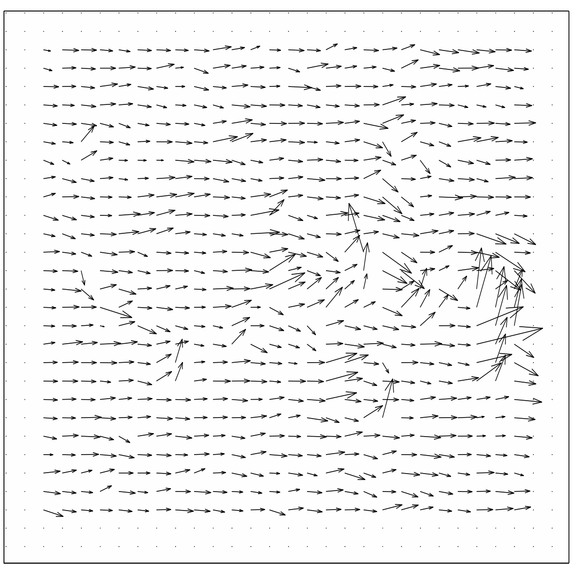
Table 2: Motion estimation results for the second sequence - Diverging Tree



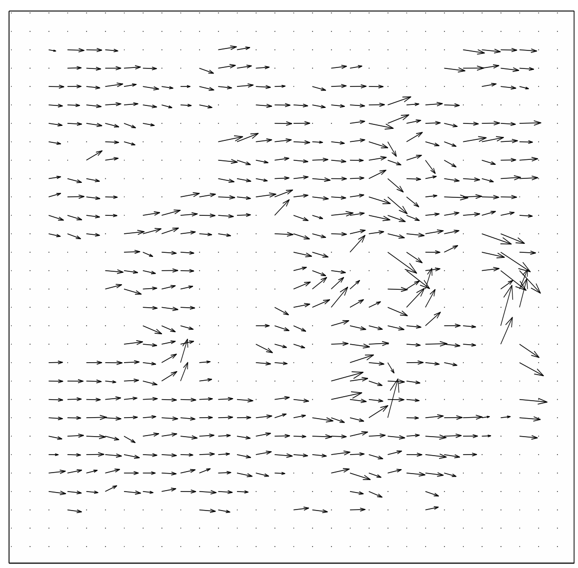
(a)



(b)

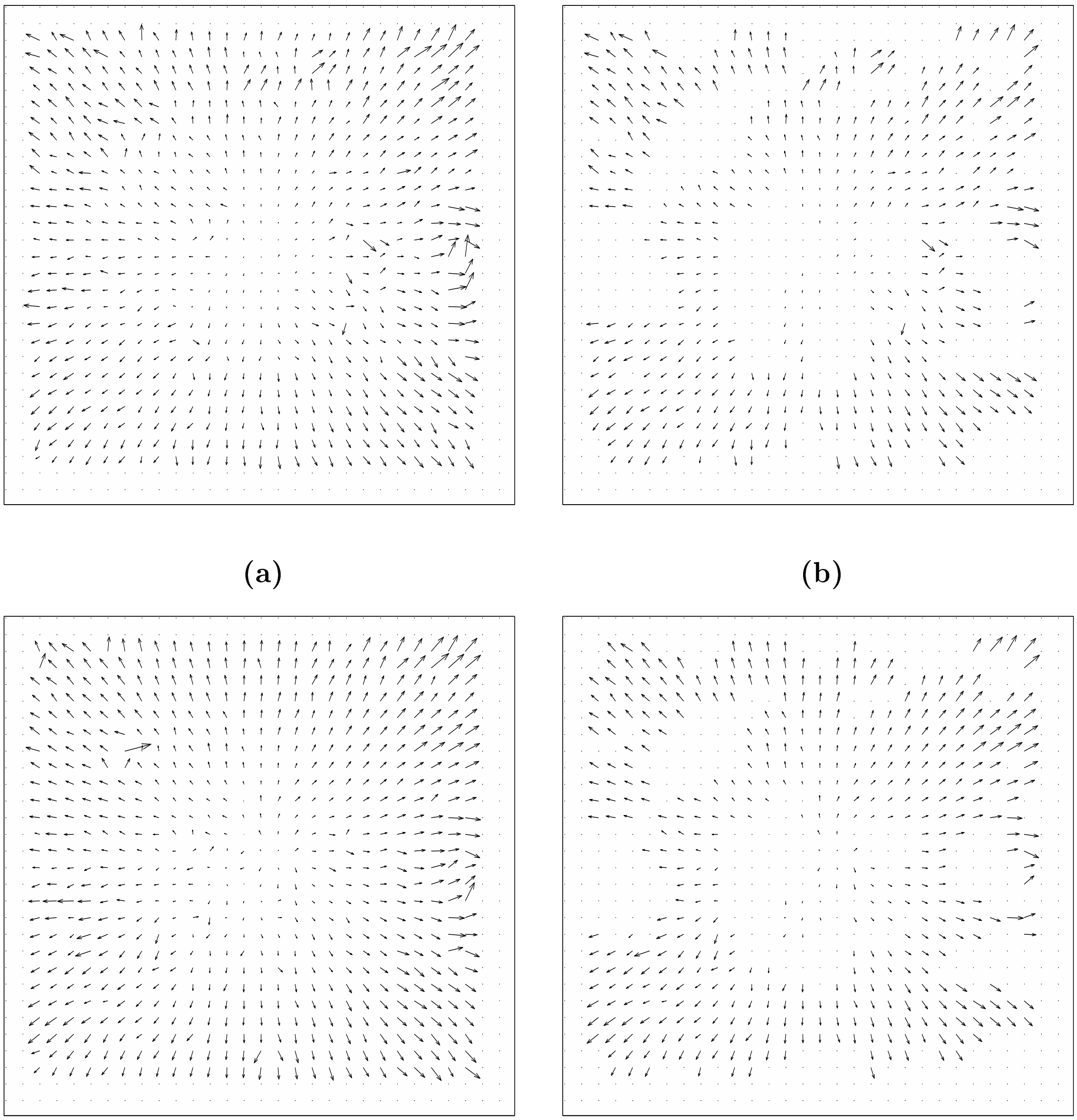


(c)



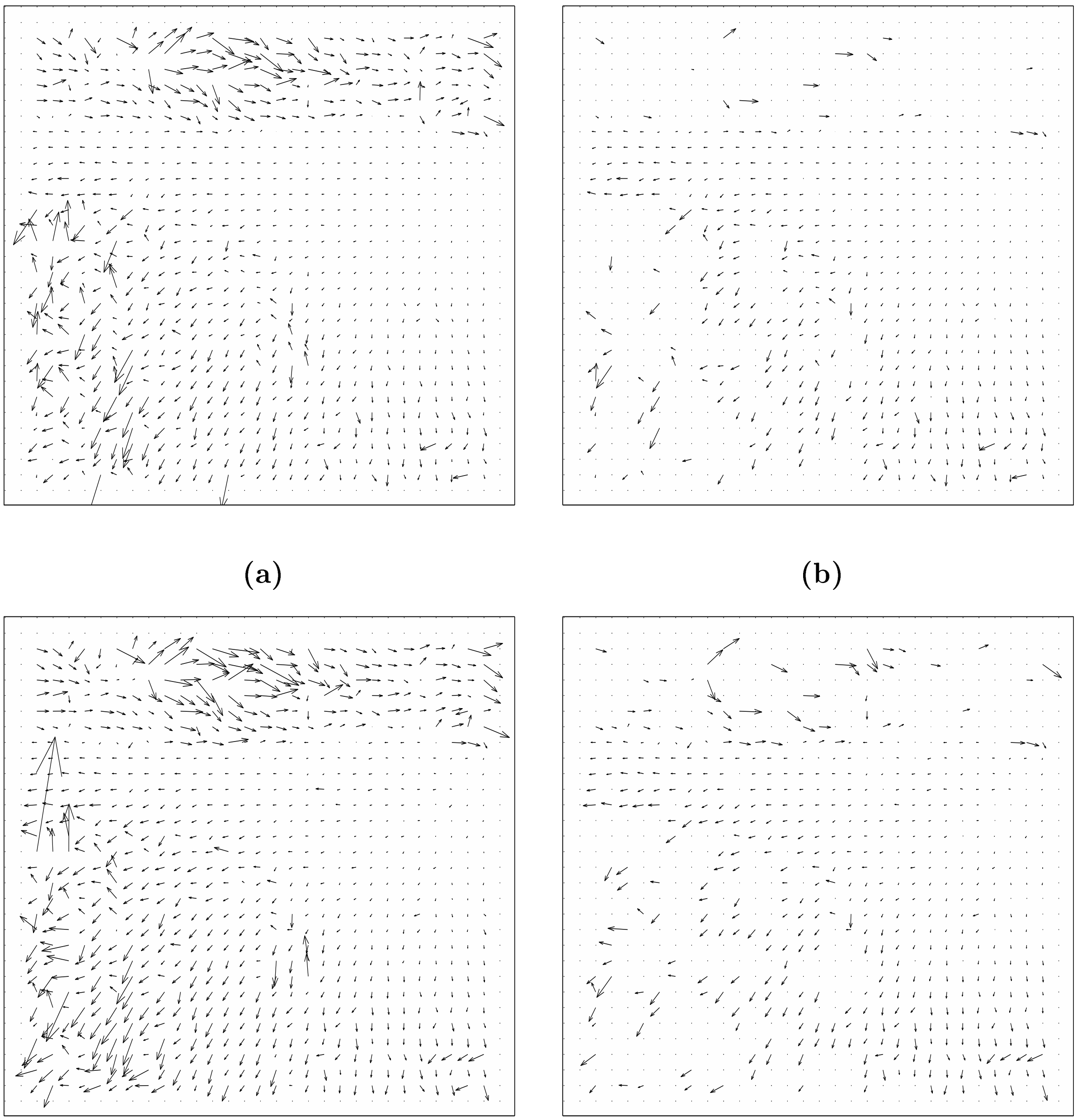
(d)

Figure 19: The first sequence (Translating Tree) results: (a) Full density estimated OF with Barron's filters, (b) Half density estimated OF with Barron's filter, (c) and (d) - similar to (a) and (b) with the optimal filters.



(a) **(b)** **(c)** **(d)**

Figure 20: The second sequence (Diverging Tree) results: (a) Full density estimated OF with Barron's filters, (b) Half density estimated OF with Barron's filter, (c) and (d) - similar to (a) and (b) with the optimal filters.



(a) **(b)** **(c)** **(d)**

Figure 21: The third (Yosemite) sequence results: (a) Full density estimated OF with Barron's filters, (b) Half density estimated OF with Barron's filter, (c) and (d) - similar to (a) and (b) with the optimal filters.

	Barron's Filters			Optimal Filters		
Optical Flow Density	Average Angular Error [degrees]	STD of Angular Error [degrees]	Mean Squared Error [pixels]	Average Angular Error [degrees]	STD of Angular Error [degrees]	Mean Squared Error [pixels]
0.15	10.7228	11.6904	0.6367	10.5244	10.1890	0.6265
0.30	11.8658	13.5434	0.7551	11.9319	12.5783	0.8151
0.45	14.2058	16.2184	1.0319	14.3679	15.0300	1.0882
0.60	17.6665	18.9380	1.3695	16.8539	16.7064	1.4228
0.75	20.9850	21.0558	1.7218	19.6586	19.1460	1.7843

Table 3: Motion estimation results for the third sequence - Yosemite (full density flow)

From the above results we can conclude several things:

1. The confidence measure that we have used is the one proposed by Barron [3]. We can see that, indeed, when the density is higher the errors increase monotonically. This property exists for all three sequences, and both for Barron's and the optimal filters. The meaning of such behavior is that the confidence measure assigns correct values for the estimated motion field.

2. The use of the optimal filters, instead on Barron's filters, yields almost for all cases better estimation results. This is certainly true for the first and the second sequences.

3. The two filter options give comparable results for the Yosemite sequence. This result can be explained by looking at the motion field histogram per each axis. It turns out that the majority of the pixels have very small motion vector, which Barron's filters are near optimal for. In order to better understand this behavior, we have computed the estimation errors for a 100×100 pixels block, taken from the lower left part of the image. This region corresponds to very high (in the norm sense) motion vectors. As

Table 4 shows, for this part of the image, the optimal filters are much better suited. In any case, note that by using the correct prior for the motion probability in the design procedure, the optimal filters results can obtain a better performance.

	Barron's Filters			Optimal Filters		
Optical Flow Density	Average Angular Error [degrees]	STD of Angular Error [degrees]	Mean Squared Error [pixels]	Average Angular Error [degrees]	STD of Angular Error [degrees]	Mean Squared Error [pixels]
0.1	20.1984	15.2986	1.3412	15.1797	13.2997	1.1725
0.2	20.6894	16.6489	1.4479	15.2730	13.4365	1.1550
0.3	21.4840	18.1293	1.5613	15.6909	13.9305	1.2070
0.4	23.4377	20.6302	1.7168	16.2974	14.4202	1.2836
0.5	24.6017	22.1826	1.8251	17.0127	15.2089	1.3603
0.6	26.2523	23.5834	1.9709	18.1218	16.6676	1.5159
0.7	27.8886	25.0859	2.1615	19.3029	18.3003	1.7011
0.8	29.2352	26.0454	2.3639	20.7312	20.0231	1.9331
0.9	30.2924	26.5315	2.5123	21.5139	20.7766	2.1143
1.0	30.4914	26.3647	2.9568	22.4840	21.8988	2.6406

Table 4: Motion estimation results for the third sequence - Yosemite - Only the lower left part of the mage of size 100×100 pixels

5 Summary and Conclusions

In this paper we have proposed a new design procedure for the filters which are required in gradient based motion estimation algorithms. The proposed design procedure generates a set of optimal filters - minimizing a penalty which was shown to be closely related to the motion estimation error. There are several benefits to the new obtained optimal filters. Their design procedure can takes into account:

1. The images spectrum, in order to suggest the best matching filters for these images.

2. The transformation prior, in order to match the characteristics of the expected motion.

3. The available number of taps.

Since we have been dealing with the BCCE throughout this paper, the obtained filters can be applied with successful results for all the motion estimation algorithms which uses the BCCE as the base-line. Simulation results, both for 1-D and for 2-D validate the optimality of the filters. In our simulations we have checked the performance of Lucas and Kanade's [14] (1-D and 2-D) and Horn and Schunck's [4] (only for 1-D) algorithms. We have tested Barron's [3], Simoncelli's [8] and the optimal filters.

In the context of the proposed optimal filters, there are many issues that should be further analyzed. The following is a partial list of them:

1. For the 2-D case, we have assumed that the optimal filters should be separable. Further research is required in order to establish conditions under-which such assumption is true.

2. The most general optical flow problem is the 3-D case, where volume of images is given. A generalization of the 1-D results to the 3-D is required. Similar to the 2-D case, it is also possible to search for separability of the filters.

3. We could use the Taylor expansion with higher derivative terms. The alternative BCCE in this case would be, for example:

$$m(x + \tau) = h(x) + g(x) \cdot \tau + f(x) \cdot \tau^2 \tag{29}$$

This off-course complicates the underlying estimation algorithms, but with potentially much smaller estimation errors. The methodology presented here can be the basis for the design of higher number of filters, in the same manner.

4. Instead of the Taylor expansion, we can use different expansion such as Fourier series. This way we get the phase based motion estimation algorithms. The alternative BCCE in this case would be, for example:

$$m(x + \tau) = h(x) + g(x) \cdot \sin(\tau) + f(x) \cdot \cos(\tau) \tag{30}$$

5. We have assumed that the local behavior of the motion flow is pure translation. We can replace this assumption with more accurate model of affine motion (translation and scaling in the 1-D case). The optimization algorithms in such a case becomes more complicated, but the resulting filters should be better suited for practical problems.

A The Choice of the Interpolating Function

Based on the assumption that the interpolating function is $b(x) = sin(\pi x)/(\pi x)$, we will show that the following equation holds:

$$\begin{aligned}
\Gamma(\mathbf{m}, \mathbf{h}, \mathbf{g}) &= \int_{x=-\infty}^{\infty} \int_{\tau=-D}^{D} \epsilon^2(x,\tau) dx d\tau = \\
&= \int_{x=-\infty}^{\infty} \int_{\tau=-D}^{D} \left[I_1(k) * \left(\hat{m}(x+\tau) - \hat{h}(x) - \hat{g}(x)\tau \right) \right]^2 dx d\tau \\
&= \sum_{k=-\infty}^{\infty} \int_{\tau=-D}^{D} \left[I_1(k) * (m(k+\tau) - h(k) - g(k)\tau) \right]^2 d\tau
\end{aligned} \tag{A-1}$$

Based on Parsevals theorem we can replace the error term $\epsilon(x,\tau)$ with its continuous Fourier transform (denoted by $\Im_\omega \left[\epsilon(x,\tau) \right]$), and sum it over the continuous frequency ω in the range $(-\infty, \infty)$:

$$\begin{aligned}
\Gamma(\mathbf{m}, \mathbf{h}, \mathbf{g}) &= \int_{\omega=-\infty}^{\infty} \int_{\tau=-D}^{D} \Im_\omega \left[\epsilon(x,\tau) \right]^2 d\omega d\tau = \\
&= \int_{\omega=-\infty}^{\infty} \int_{\tau=-D}^{D} |I_1(\omega)|^2 \left| \left(e^{j\omega\tau} \hat{M}(\omega) - \hat{H}(\omega) - \hat{G}(\omega)\tau \right) \right|^2 d\omega d\tau
\end{aligned} \tag{A-2}$$

Note that even though $I_1(k)$ is discrete time signal, we use its continuous Fourier transform Using the definition of $\hat{m}(x), \hat{h}(x), \hat{g}(x)$ as given in Equation 13, their continuous Fourier transform is given by:

$$\begin{aligned}
\hat{M}(\omega) &= \int_{-\infty}^{\infty} \hat{m}(x) e^{-j\omega x} dx = \int_{-\infty}^{\infty} \sum_{k=-L}^{k=L} m(k) b(x-k) e^{-j\omega x} dx \\
&= \sum_{k=-L}^{k=L} m(k) e^{-j\omega k} \int_{-\infty}^{\infty} b(x) e^{-j\omega x} dx = B(\omega) \sum_{k=-L}^{k=L} m(k) e^{-j\omega k} = B(\omega) M(\omega)
\end{aligned} \tag{A-3}$$

where we have defined $M(\theta) = \sum_{k=-L}^{k=L} m(k)e^{-j\theta k}$. Similar relations can be written for the filters $\hat{H}(\omega)$ and $\hat{G}(\omega)$. According to the above assumption on the interpolating function we get:

$$b(x) = \frac{sin(\pi x)}{(\pi x)} \qquad \Rightarrow \qquad B(\omega) = \begin{cases} 1 & -\pi \leq \omega \leq \pi \\ 0 & \text{otherwise} \end{cases} \tag{A-4}$$

Putting the above transforms in the error term integral obtained in A-2 we get that the frequency integral are in the range $[-\pi, \pi]$, since:

$$\begin{aligned} \Gamma(\mathbf{m}, \mathbf{h}, \mathbf{g}) &= \int_{\omega=-\infty}^{\infty} \int_{\tau=-D}^{D} |I_1(\omega)|^2 \left| \left(e^{j\omega\tau} \hat{M}(\omega) - \hat{H}(\omega) - \hat{G}(\omega)\tau \right) \right|^2 d\omega d\tau = \\ &= \int_{\omega=-\infty}^{\infty} \int_{\tau=-D}^{D} |I_1(\omega)|^2 |B(\omega)|^2 \left| \left(e^{j\omega\tau} M(\omega) - H(\omega) - G(\omega)\tau \right) \right|^2 d\omega d\tau = \\ &= \int_{\omega=-\pi}^{\pi} \int_{\tau=-D}^{D} |I_1(\omega)|^2 \left| \left(e^{j\omega\tau} M(\omega) - H(\omega) - G(\omega)\tau \right) \right|^2 d\omega d\tau = \end{aligned} \tag{A-5}$$

Using again Parsevals theorem we get:

$$\begin{aligned} \Gamma(\mathbf{m}, \mathbf{h}, \mathbf{g}) &= \int_{\omega=-\pi}^{\pi} \int_{\tau=-D}^{D} |I_1(\omega)|^2 \left| \left(e^{j\omega\tau} M(\omega) - H(\omega) - G(\omega)\tau \right) \right|^2 d\omega d\tau \\ &= \sum_{k=-\infty}^{\infty} \int_{\tau=-D}^{D} \left[I_1(k) * (m(k+\tau) - h(k) - g(k)\tau) \right]^2 d\tau \end{aligned} \tag{A-6}$$

which proves equation A-1. $\square$

B The Quadratic Form of the Optimization Problem

This appendix describes in details the error term as described in section 3.1. Let us define the weight function:

$$W(\theta) = \alpha + (1-\alpha)|I_1(\theta)|^2 \tag{B-1}$$

The error term given in the frequency domain can be presented by:

$$\Gamma_{\text{total}}(\mathbf{m}, \mathbf{h}, \mathbf{g}) = \int_{\theta=-\pi}^{\pi} \int_{\tau=-D}^{D} W(\theta) \left| \left(e^{j\theta\tau} \hat{M}(\theta) - \hat{H}(\theta) - \hat{G}(\theta)\tau \right) \right|^2 d\theta d\tau \tag{B-2}$$

Using the fact that $\hat{M}(\theta)$, $\hat{H}(\theta)$, and $\hat{G}(\theta)$ are the DFT of $\hat{m}(k)$, $\hat{h}(k)$, and $\hat{g}(k)$, respectively, we get:

$$\Gamma_{\text{total}}(\mathbf{m}, \mathbf{h}, \mathbf{g}) = \int_{\theta=-\pi}^{\pi} \int_{\tau=-D}^{D} W(\theta) \left| \left[e^{j\theta\tau} \mathbf{f}(\theta),\ -\mathbf{f}(\theta),\ -\tau \mathbf{f}(\theta) \right] \mathbf{x} \right|^2 d\theta d\tau \tag{B-3}$$

where $\mathbf{f}(\theta)$ is defined as the Discrete Fourier Transform (DFT) vector:

$$\mathbf{f}(\theta) = \left[e^{jL\theta}, ..., e^{-jL\theta} \right] \tag{B-4}$$

and the vector $\mathbf{x}$ consist of all the discrete filters coefficients, namely, $\mathbf{x^T} = [\mathbf{m^T}, \mathbf{h^T}, \mathbf{g^T}]$. Defining the Hermite matrix $\mathbf{F}(\theta)$ to be:

$$\mathbf{F}(\theta) = \mathbf{f}^H(\theta)\mathbf{f}(\theta) = \begin{bmatrix} 1 & e^{-j\theta} & \dots & e^{-j\theta 2L} \\ e^{j\theta} & 1 & \dots & e^{-j\theta(2L-1)} \\ . & . & & . \\ e^{j\theta 2L} & e^{j\theta(2L-1)} & \dots & 1 \end{bmatrix} \tag{B-5}$$

the above error term can be rewritten as:

$$\begin{aligned} &\Gamma_{\text{total}}(\mathbf{m}, \mathbf{h}, \mathbf{g}) = \qquad \text{(B-6)} \\ = \ &\mathbf{x}^H \left[\int_{\theta=-\pi}^{\pi} W(\theta) \int_{\tau=-D}^{D} \begin{pmatrix} \mathbf{F}(\theta) & -e^{-j\theta\tau}\mathbf{F}(\theta) & -\tau e^{-j\theta\tau}\mathbf{F}(\theta) \\ -e^{j\theta\tau}\mathbf{F}(\theta) & \mathbf{F}(\theta) & \tau\mathbf{F}(\theta) \\ -\tau e^{j\theta\tau}\mathbf{F}(\theta) & \tau\mathbf{F}(\theta) & \tau^2\mathbf{F}(\theta) \end{pmatrix} d\tau d\theta \right] \mathbf{x} = \\ = \ &\mathbf{x}^H \left[\int_{\theta=-\pi}^{\pi} W(\theta) \int_{\tau=-D}^{D} \begin{pmatrix} 1 & -e^{-j\theta\tau} & -\tau e^{-j\theta\tau} \\ -e^{j\theta\tau} & 1 & \tau \\ -\tau e^{j\theta\tau} & \tau & \tau^2 \end{pmatrix} d\tau \otimes \mathbf{F}(\theta) d\theta \right] \mathbf{x} \end{aligned}$$

The notation $\otimes$ represents the Kronecker multiplication of matrices [13]. The internal integral, which corresponds to τ, can be computed analytically. The result is the following expression:

$$\begin{aligned} \Gamma_{\text{total}}(\mathbf{m}, \mathbf{h}, \mathbf{g}) = \mathbf{x}^H \Bigg[\int_{\theta=-\pi}^{\pi} W(\theta) \cdot \qquad \text{(B-7)} \\ \begin{pmatrix} 2D & -2\frac{sin(\theta D)}{\theta} & -2j\frac{\theta D cos(\theta D) - sin(\theta D)}{\theta^2} \\ -2\frac{sin(\theta D)}{\theta} & 2D & 0 \\ 2j\frac{\theta D cos(\theta D) - sin(\theta D)}{\theta^2} & 0 & \frac{2\,D^3}{3} \end{pmatrix} \otimes \mathbf{F}(\theta) d\theta \Bigg] \mathbf{x} \end{aligned}$$

By defining the following Positive Definite (PD) matrix:

$$\mathbf{R} = \int_{-\pi}^{\pi} W(\theta) \begin{pmatrix} 2D & -2\frac{sin(\theta D)}{\theta} & -2j\frac{\theta D cos(\theta D) - sin(\theta D)}{\theta^2} \\ -2\frac{sin(\theta D)}{\theta} & 2D & 0 \\ 2j\frac{\theta D cos(\theta D) - sin(\theta D)}{\theta^2} & 0 & \frac{2\,D^3}{3} \end{pmatrix} \otimes \mathbf{F}(\theta) d\theta \tag{B-8}$$

we see that, according to our notations in section 3.1, the error term can be represented as a quadratic form $\Gamma_{\text{total}}(\mathbf{m}, \mathbf{h}, \mathbf{g}) = \mathbf{x}^H \mathbf{R} \mathbf{x}$. $\square$

In the case where the filters $\mathbf{m}$ and $\mathbf{h}$ are identical (as in Simoncelli's case), using similar steps as described above we get that $\Gamma_{\text{total}}(\mathbf{m}, \mathbf{g})$ is given by (see equation B-2):

$$\begin{aligned} \Gamma_{\text{total}}(\mathbf{m}, \mathbf{g}) &= \int_{\theta=-\pi}^{\pi} \int_{\tau=-D}^{D} W(\theta) \left| \left(\left[e^{j\theta\tau} - 1 \right] \hat{M}(\theta) - \hat{G}(\theta)\tau \right) \right|^2 d\theta d\tau \qquad \text{(B-9)} \\ \Rightarrow \mathbf{R} &= \int_{-\pi}^{\pi} W(\theta) \begin{pmatrix} 4D - 4\frac{sin(\theta D)}{\theta} & \frac{2j[\theta D cos(\theta D) + sin(\theta D)]}{\theta^2} \\ \frac{2j[\theta D cos(\theta D) - sin(\theta D)]}{\theta^2} & \frac{2D^3}{3} \end{pmatrix} \otimes \mathbf{F}(\theta) d\theta \end{aligned}$$

C Asymptotic Optimal Filters

Assume that $D \to 0$ (very small motion vectors), $\alpha = 1$ (which means that the weights are 1 for all frequencies), and that $b(x) = sinc(x) = \frac{sin(\pi x)}{\pi x}$. The question is: What are the optimal 1-D filters $\mathbf{m}$, $\mathbf{h}$ and $\mathbf{g}$ in this case?

The optimal filters $\mathbf{m}$, $\mathbf{h}$ and $\mathbf{g}$ should minimize the following penalty (see equation 17):

$$\Gamma(\mathbf{m}, \mathbf{h}, \mathbf{g}) = \sum_{k=-\infty}^{\infty} \int_{\tau=-D}^{D} \left[\hat{m}(k+\tau) - \hat{h}(k) - \hat{g}(k)\tau\right]^2 d\tau \tag{C-1}$$

The above expression contains only one difference compared to the original Equation 17 - $I_1(k)$ is omitted since $\alpha = 1$. Since $D \to 0$ we get that the following is a good approximation relation:

$$\text{for} - D \leq \tau \leq D, \qquad \hat{m}(k+\tau) = \hat{m}(k) + \tau \cdot \hat{m}'(k) + o\{\tau^2 \cdot \hat{m}''(k)\} \tag{C-2}$$

Replacing these terms in equation C-1 we get:

$$\Gamma(\mathbf{m}, \mathbf{h}, \mathbf{g}) = \sum_{k=-\infty}^{\infty} \int_{\tau=-D}^{D} \left[\hat{m}(k) - \hat{h}(k) + \hat{m}'(k)\tau - \hat{g}(k)\tau + o\{\tau^2 \cdot \hat{m}''(k)\}\right]^2 d\tau \tag{C-3}$$

Since the range of the integral over τ is $[-D, D]$ and D is assumed to very small, the first and the second differences (i.e. $\left[\hat{m}(k) - \hat{h}(k)\right]$ and $[\hat{m}'(k) - \hat{g}(k)]\,\tau$) are the dominant terms within the integral. Zeroing these differences is thus the optimal solution. Since we have to zero these terms for each τ, minimum for this penalty is given by the following set of conditions:

1. $\hat{m}(k) = \hat{h}(k)$,
2. $\hat{g}(k) = \hat{m}'(k)$,
3. $\hat{m}(k)$ should be the non-trivial minimizer or the penalty - $\sum_{k=-\infty}^{\infty} \hat{m}''(k)^2$. This is equivalent to a penalty for non-smoothness of the filter $\hat{m}(k)$; we require the filter $\hat{m}(k)$ to be as flat as possible.

Requiring that the interpolated filters $\hat{m}(k)$ and $\hat{h}(k)$ should be equal is equivalent to the requirement $\mathbf{m} = \mathbf{h}$. According to the above,the filter $\mathbf{g}$ is given by the sampled derivative of the interpolated filter $\mathbf{m}$:

$$b'(k) = \begin{cases} 0 & k = 0 \\ \frac{(-1)^k}{k} & k \neq 0 \end{cases} \quad \Rightarrow \quad g(j) = \sum_{k=-L}^{L} m(k) \frac{(-1)^{j-k}}{j-k}, \qquad -L \leq j \leq L \qquad \text{(C-4)}$$

Regarding the third condition, we use the fact that:

$$b''(x) = \begin{cases} \frac{-\pi^2}{3} & x = 0 \\ \frac{-2(-1)^x}{x^2} & x \neq 0 \end{cases} \qquad \text{(C-5)}$$

The penalty of $\sum_{k=-\infty}^{\infty} m''(k)^2$ is given by:

$$\Gamma(\mathbf{m}) = \sum_{k=-\infty}^{\infty} m''(k)^2 \;=\; \sum_{k=-\infty}^{\infty} \left[\sum_j m(j) b''(k-j) \right]^2 = \qquad \text{(C-6)}$$

$$
\begin{aligned}
&= \sum_{j_1}\sum_{j_2} m(j_1)m(j_2)\sum_{k=-\infty}^{\infty} b''(k-j_1)b''(k-j_2) \\
&= \sum_{j_1}\sum_{j_2} m(j_1)m(j_2)C(j_1-j_2)
\end{aligned}
$$

where we have defined the autocorrelation sequence $C(j_1-j_2) = \sum_{k=-\infty}^{\infty} b''(k-j_1)b''(k-j_2)$. The above penalty can be rewritten in matrix-vector form:

$$
\Gamma(\mathbf{m}) = \mathbf{m}^T\mathbf{C}\mathbf{m} = \mathbf{m}^T \begin{bmatrix} C(0) & C(1) & .. & C(2L) \\ C(1) & C(0) & .. & C(2L-1) \\ . & . & . & . \\ C(2L) & C(2L-1) & .. & C(0) \end{bmatrix} \mathbf{m} \qquad \text{(C-7)}
$$

The matrix $\mathbf{C}$ is symmetric Toeplitz matrix which decays very fast ($1/k^4$) around the main diagonal. Results show that the obtained filter $\mathbf{m}$ can be approximated by a raised cosine function:

$$
m(k) \approx 1 + \cos\left[\frac{\pi k}{L+1}\right] \qquad -L \leq k \leq L \qquad \text{(C-8)}
$$

as described in sub-section 3.2. □

D References

[1] A. Singh, *Optic Flow Computation: A unified Perspective.* IEEE Computer Society Press, California, 1991. 1-st Edition.

[2] B. Jahne, *Digital Image Processing - Concepts, Algorithms, and Scientific Applications.* Springer-Verlag, 1995. 3-st Edition.

[3] J. L. Barron, D. J. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International Journal of Comuter Vision*, vol. 12, pp. 43–77, 1994.

[4] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Inteligence*, vol. 17, pp. 185–203, 1981.

[5] T. Chin, W. Karl, and A. Willsky, "Probabilistic and sequential computation of optical flow using temporal coherence," *IEEE Trans. Image Processing*, vol. 3, pp. 773– 788, 1994.

[6] D. J. Fleet and K. Langley, "recursive filters for optical flow," *IEEE Trans. Pattern Analysis and Machine Inteligence (PAMI)*, vol. 17, pp. 61–67, 1995.

[7] M. Elad and A. Feuer, "Recursive optical flow estimation - adaptive filtering approach," *Submitted to the Journal of Visual Comm. and Image Represetation*, 1996.

[8] E. P. Simoncelli, "Design of multi-dimensional derivatives filters," in *The IEEE International Conf. on Image Processing, Austin Tx*, 1994.

[9] R. Anderson and P. Bloomfield, "Numerical differentiation procedures for nonexact data," *Numerical Mathematics*, vol. 22, pp. 157–182, 1974.

[10] B. Porat, *A Course in Digital Signal Processing.* Wiley, 1997. 1-st Edition.

[11] E. Simoncelli, W. Freeman, E. Adelson, and D. Heeger, "Shiftable multiscale transforms," *IEEE Trans. Information Theory*, vol. 38, pp. 587–607, 1992.

[12] Y. Hel-Or and P. Teo, "A common framework for steerability, motion estimation and invariant feature detection," Tech. Rep. STAN-CS-TN-96-28, Stanford University, 1996.

[13] R. A. Horn and C. J. Johnson, *Matrix Analysis.* Cambridge University Press, 1985. 1-st Edition.

[14] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. DARPA, Image Understanding Workshop*, pp. 121–130, 1981.