On the Design of Filters For Gradient-Based Motion Estimation*

Michael Elad† Patrick Teo‡ Yacov Hel-Or§

4 October, 2004

Abstract

Gradient based approaches in motion estimation (Optical–Flow) refer to those techniques that estimate the motion of an image sequence based on local derivatives in the image intensity. In order to best evaluate local changes, specific filters are applied to the image sequence. These filters are typically composed of a spatiotemporal pre-smoothing filter followed by discrete derivative ones. The design of these filters plays an important role in the estimation accuracy. This paper proposes a method for such a design. Unlike previous methods that consider these filters as optimized approximations for continuum derivatives, the proposed design procedure defines the optimality directly with respect to the motion estimation goal. One possible result of the suggested scheme is a set of image dependent filters that can be computed prior to the estimation process. An alternative interpretation is the creation of generic filters, capable of treating natural images. Simulations demonstrate the validity of the new design approach.

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

1 Introduction

Estimating motion between two images plays a vital role in many applications, 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. In the work by Barron et. al. [3] a comparative survey on many motion estimation algorithms is given, considering different disciplines. One family of such algorithms which is found to perform very well is the gradient-based methods, originally proposed by Horn and Schunck [4].

*This work was done while the authors where with Hewlett-Packard laboratories - Israel.

†Corresponding Author: Department of Computer Science, The Technion, Haifa, 32000 Israel. Email: elad@cs.technion.ac.il, Phone:+972-4-829-4169, Fax: +972-4-829-3900.

‡Shutterfly, Redwood City, CA.

§Department of Computer Science, Inter-Disciplinary Center, Herzliya, 46150 Israel.

The gradient-based methods emerge all 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 so called *brightness change constraint equation* (BCCE) [1, 2, 3, 4]. Denoting the image sequence gray values at the spatial location (x, y) and time t by a function $I(x, y, t)$, the constant brightness assumption on two consecutive images implies

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

Taking the first order Taylor expansion of the right term around the spatiotemporal point (x, y, t) and neglecting higher order terms gives

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

Defining $(u^x, u^y) = (\delta x/\delta t, \delta y/\delta t)$, this vector specifies the spatial velocity of each point in the image sequence. Assuming a time difference $\delta t = 1$, the above equation can be rewritten as

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

where I_x, I_y and I_t denote the spatial and temporal derivatives, correspondingly. This equation, known as the BCCE, relates the spatial and temporal derivatives 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 information is required in order to uniquely recover the motion field. For this purpose, a spatial smoothness [1, 2, 3] and/or temporal one [1, 5, 6, 7] on the motion field can be imposed. Indeed, the main difference between various gradient based methods lays in the way these smoothness assumptions are applied.

One issue which is critical to the application of the above BCCE is the fact that image gradients are computed based on sampled information. It is commonly agreed that approximating the spatiotemporal gradients by finite differences produces error in the above equation and subsequently in the estimated motion [3, 8]. One of the major conclusions in the survey work in [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"* .

Beyond the need for gradient estimation, it is commonly recommended to apply spatiotemporal pre-smoothing to the image sequence [1, 2, 3]. There are several reasons for this pre-processing stage:

1. Additive noise is amplified when combined with gradient operations. Pre-smoothing attenuates this noise and thus reduces its damaging effects;

2. Pre-smoothing approximates the data to locally behave like tilted planes. Such behavior justifies the first order Taylor approximation of the BCCE as described above;

3. Pre-smoothing attenuates the spatial and temporal aliasing effects in the image sequence. This way the estimated motion is more accurate; and

4. Typical finite differentiators are more accurate at low frequencies. Since pre-smoothing reduces the high frequencies we get that the overall gradient estimation becomes more accurate.

Since the pre-smoothing and the gradient approximations are both linear and spatiotemporal invariant (LSTI) operations, it is possible to combine them into a single filtering stage. In the most general case, it is suggested to implement the BCCE 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]. These methods treat the above question as a problem of approximating the continuum derivative operators, overlooking the fact that these filters are only the means for the purpose of motion estimation. The question that is addressed in this paper refers to the design of these filters directly with respect to the goal in mind. The undertaken approach suggests finding the filters which are (near–) optimal directly with respect to the motion estimation error.

A somewhat similar attempt using a different methodology is described in [9], where several gradient operators are compared by studying their performance on the shape from motion (SFM) task. While we attach the design of the derivative filters to a specific task, we take a different path of optimizing these filters rather than comparing known methods.

Since 3-D separable filters are easier to implement, it is commonly assumed that F_1, F_2 and F_3 are separable [3, 4, 8]. We have chosen to apply this line of reasoning in this paper too, namely, the designed filters are 1-D kernels which are used to construct the 3-D required kernels (similar to [8]). However, the methodology described here can be extended directly to higher dimensions in a straightforward manner.

This paper is organized as follows: Section 2 briefly surveys the existing approaches for the design of the required filters. Section 3 presents our proposed approach. In section 4 several examples illustrating the new design procedure are given. Summary and conclusions are given in section 5. The interested reader is directed to [10] for a shorter version of this work with a different perspective.

2 Existing Motion Estimation Filters

2.1 General

The numerical analysis literature contains many methods for approximating differentiation [2, 11]. Most of the papers describing optical flow estimation through the BCCE either apply trivial differentiations like $\frac{1}{2}[-1, 0, 1]$ [4], or more sophisticated alternatives [3]. In some papers the choice of these filters is even not mentioned, hinting to the hidden belief that these filters have small effect on the performance obtained.

In their original paper, Horn and Schunck proposed an approximation of the gradient filters with no distinct pre-smoothing [4]. The proposed gradient is obtained by averaging the four first differences taken over a $2\times2\times2$ cube in the image sequence (see Figure 1). The obtained derivatives refer to the center point of the processed cube. No motivation or justification for this choice of gradient estimation is given. In [3], these filters were said to be *"relatively crude form of numerical differentiation and can be the source of considerable error"*.

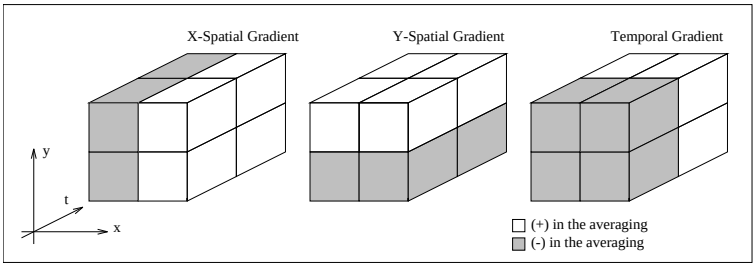


Figure 1: Horn and Schunck's proposed filters.

Barron et. al. proposed as an alternative the application of a $5 \times 5 \times 5$ spatiotemporal presmoother, constructed as a sampled Gaussian filter with 1.5 variance at each axis [3]. This value of variance was found empirically to give the best results. The differentiator proposed is the 5-tap 1-D filter $\frac{1}{12}[-1, 8, 0, -8, 1]$, which is a result of a design procedure described in [2]. For completeness, the next sub-section shortly describes this and related design procedures.

2.2 Optimal Differentiators

Assume that the required differentiator is a linear phase finite impulse response (FIR) filter with odd number of taps. These assumptions are reasonable because (i) linear phase filters assures a constant phase delay, which means a preservation of the signal structure [12]; (ii) linear phase can only be implemented by FIR filters [12]. This is of-course a benefit since FIR filters are easily implemented to images; and (iii) a filter with even number of taps approximates the gradient at the midpoint between two adjacent pixels, whereas a filter with odd number of taps corresponds the calculated gradients to the pixels' center.

This choice of differentiator implies that its impulse response is antisymmetric [12]. The general form of the filter's impulse response is given by

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

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

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

It is easily shown from the above equation that using odd number of taps constrains this transfer function to be zero at the frequencies $\theta = \pm\pi$ [12]. This in turn means that the approximated differentiation is severely compromised at high frequencies. However, this error is negligible if we assume that high frequencies are attenuated by the pre-smoothing operation.

Our goal is to obtain a near-accurate differentiator transfer function $D(\theta) = j\theta$. The filter coefficients $\{d(k)\}_{k=1}^{L}$ are to be designed to meet this requirement as closely as possible. There are several ways to define an optimality criteria 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) criteria, and the minimax Remez exchange algorithm [12].

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

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

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

Figure 2 presents the 3-D filters that are applied using the above described filters. A 3-D $[5\times5\times5]$ pre-smoothing kernel is first applied to the image sequence, followed by a differentiator per each axis separately. The resulting three 3-D kernels F_1, F_2, and F_3, refer to the BCCE as given in Equation 4.

Figure 3 presents the actual 1-D filters for 9-taps: pre-smoother $p(x)$ and differentiator (combined with the smoother) $d(x) * p(x)$. The pre-smoother is taken to be a sampled Gaussian filter with 1.5 variance, and the differentiator is the optimal 5-taps filter as described above. The presented filters are given in their continuous form, interpolated using the sinc function. Figure 4 presents

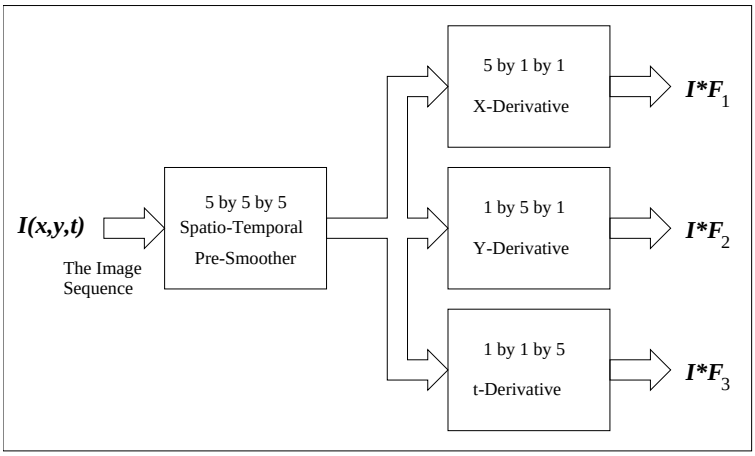
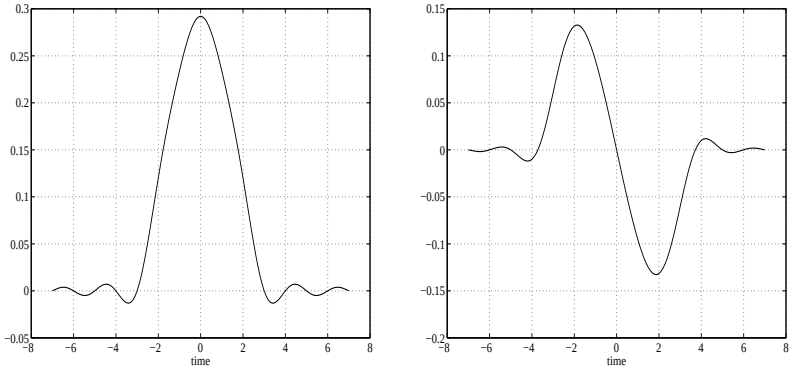


Figure 2: Barron's proposed filters configuration

the power spectrum of the differentiator, $|D(\theta)P(\theta)|$, compared to an analytic differentiation of the smoother filter, $|j\theta \cdot P(\theta)|$. These filters were proposed by Barron [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 differentiator suggested by Barron

2.3 Simoncelli's Filters

An innovative design approach for the determination of the required filters is presented by Simoncelli [8]. Several important aspects that were traditionally overlooked are referred in this work:

1. **Continuum versus the discrete grid** - Most of the differentiators are designed on the given discrete grid. However, the given discrete signal is a result of a 3-D sampling of an originally continuous image sequence. In order to be more accurate, the design of discrete differentiator

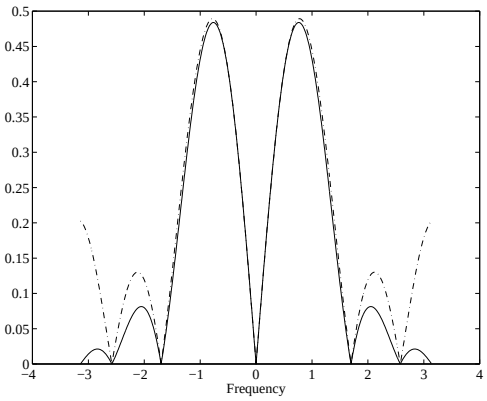


Figure 4: Barron's differentiator filters in the frequency domain $|D(\theta)P(\theta)|$ (solid line) and $|j\theta \cdot P(\theta)|$ (dashed line).

filters should be done with respect to the continuum. This means that a function $b(x, y, t)$, interpolating the signal to the continuum is required. Simoncelli [8] suggests using the sinc function or more *"gentler function"* such as a Gaussian of some sort.

2. **Design of pairs of filters** - Instead of designing a single derivative filter, two filters are designed simultaneously, where one of them relates to the other as a differentiator. This new idea suggests that a signal $I(x)$ and its derivative $I_x(x)$ are hard to come-by accurately. Alternatively, we can supply the pair $\{I * p\}(x)$ and $\{I * d\}(x)$ which relate to each other as an original and its derivative in a more accurate form.

   If we denote a smoothing and a differentiating 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 weigh the approximation error accordingly. Simoncelli [8] suggests that the design should be done in the frequency domain, and the weight should manifest the $1/\theta$ spectral behavior of typical images.

Using the above, Simoncelli's approach proposes a design of the filters using the WLS criterion

$$\epsilon^2(\underline{p}, \underline{d}) = \int_{-\pi}^{\pi} \frac{1}{|\theta|^{0.5}} [j\theta P(\theta) - D(\theta)]^2 d\theta \tag{8}$$

$$\text{where:} \qquad P(\theta) = \sum_{k=-L}^{L} p(k) \exp\{-jk\theta\} \quad \text{and} \quad D(\theta) = \sum_{k=-L}^{L} d(k) \exp\{-jk\theta\}.$$

The solution is obtained using the Singular Value Decomposition (SVD) approach. The resulting filters are normalized so that $p(k)$ yields a fixed DC response. This way, a pair of 1-D filters

is obtained: $p(k)$, the smoothing, and $d(k)$, the derivative filter. Referring to equation (4) and assuming the separability of the 3-D filters, the filters to be used are

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

$$\text{where:} \qquad F_1(x,y,t) = d(x) * p(y) * p(t)$$

$$F_2(x,y,t) = p(x) * d(y) * p(t)$$

$$F_3(x,y,t) = p(x) * p(y) * d(t).$$

Figure 5 presents the 3-D filters that are applied using the above described filters. This flow of operations is consistent with equation (9).

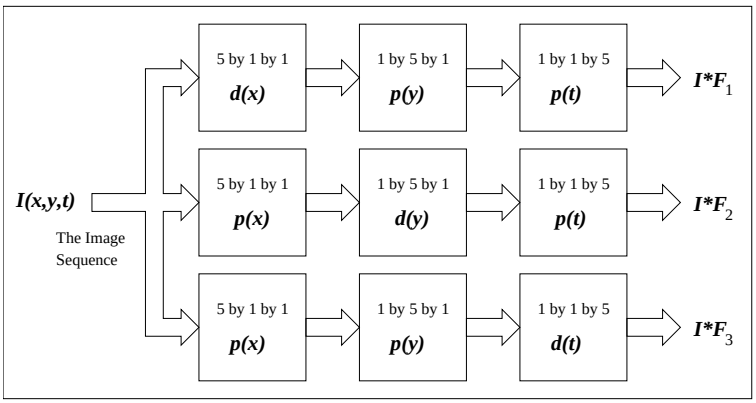
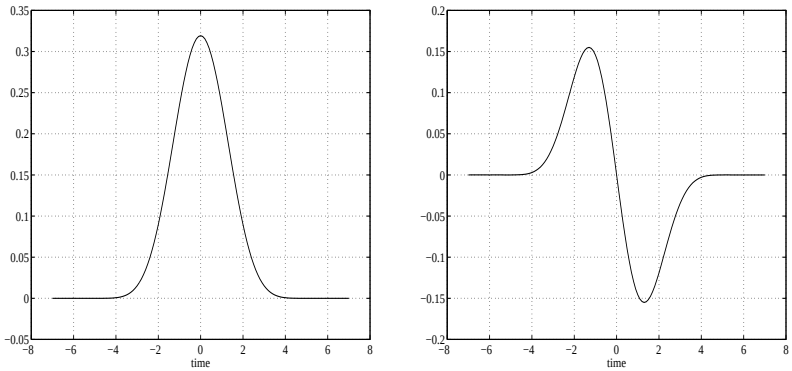


Figure 5: Simoncelli's proposed filters configuration.

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



(a) **(b)**

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

2.4 Existing Approaches - Summary

So far we have presented the existing methods for the determination of the filters to be applied in the BCCE for motion estimation. Among the various properties of these proposed filters, the following concepts are beneficial and should be kept in any design of derivative filters:

1. The produced 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 continuum.
3. The design of all the required filters should be done simultaneously, in order to assure general form of optimality.
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 for the required filters, and should exploit them in the best possible manner.
6. If possible, the design should yield separable filters which are easier to implement. However, in the following discussion we shall deal with separable filters as emerging from 1-D design procedure.

Beyond these specifications, there are two more important requirements that are missing from the discussion so far, and should be met in order to obtain better results:

1. For the general optimal solution the motion characteristics should be considered. For example, assume that the motion vectors components are in the range of $[-2, 2]$ or $[-0.2, 0.2]$. This choice should somehow influence the applied filters. No existing method suggests such property.
2. Most important of all, minimization should be done with respect to the motion estimation error (or related terms, if direct minimization of this error is too complicated), which is our final goal.

The next section presents a new approach for the design of the required filters with the attempt to answer all the above requirements.

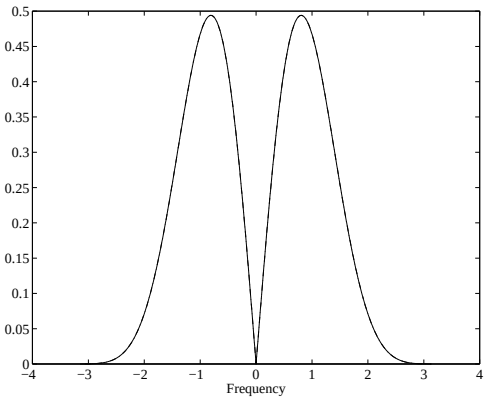


Figure 7: Simoncelli's proposed filters in the frequency domain $|D(\theta)|$ (solid line) and $|j\theta \cdot P(\theta)|$ (dashed line) (they are actually one on top of the other).

3 The Proposed Approach

3.1 Shiftable Filters

As said earlier, we will restrict our treatment to the 1-D case. Extension of the methodology presented here to higher dimensions is straightforward, but involves longer and cumbersome expressions.

Assume that a pair of 1-D discrete images (vectors) are given with apparent small motion between them. These images are denoted by $I_1(k)$ and $I_2(k) = I_1(k + \tau(k))$, $(k \in [1..L_I])$. The 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

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

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

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

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 mentioned 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 bigger ones. In the general case, however, this behavior is too restrictive. There is no reason to assume that small displacements are required more accurately than others. It is a desirable property to be able to closely control the behavior of the above error, and match it to prior knowledge on the motion field.

The above equation contains the application of 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 the general case, the application of three different filters can be suggested as an alternative. These filters will be designed to meet some desired error properties according to our prior knowledge on the motion field. This general form can be expressed as

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

where $\hat{m}(x),\ \hat{h}(x),$ and $\hat{g}(x)$ are three different filters. 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 whose interpolated versions are the continuous filters

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

where $\mathbf{m}^T = [m(-L), ..., m(L)]$, $\mathbf{h}^T = [h(-L), ..., h(L)]$, $\mathbf{g}^T = [g(-L), ..., g(L)]$, and $\mathbf{b}(\mathbf{x})^T = [b(x+L), ..., b(x-L)]$. Though the number of taps of these filters is $2L+1$, there is no restriction to an odd support in the general case.

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 error term that should be minimized is

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

where the term $\epsilon(x, \tau(x))$ is as defined in Equation (12), and $E_z\{f(z)\}$ is the expectation value of $f(z)$. The error term is minimized over all possible x and $\tau(x)$. While the minimization over all the relevant 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 will be done over the admissible range of the parameters. A simple parametric expansion using again the Taylor series gives

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

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. More complicated motion model may be treated similarly.

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

$$\hat{m}(x + \tau) * I_1(k) = \hat{h}(x) * I_1(k) + \tau \cdot \hat{g}(x) * I_1(k) + \epsilon(x, \tau). \tag{16}$$

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 [13], where a transformed version of a function is expressed as a linear sum of a set of basis functions.

3.2 The Penalty Function

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

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

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, forcing smaller error for more probable values. Similarly, if we know that errors around the origin ($\tau = 0$) are detrimental (e.g. in a pyramidal motion estimation methods), this could be easily inserted as weight to the above defined penalty function.

In order to avoid the required convolution in the above function, it is possible to apply the minimization in the frequency domain. Using Parseval's theorem we get that the design goal is equivalent to the following:

$$\begin{aligned}\Gamma(\mathbf{m},\mathbf{h},\mathbf{g}) &= \int_{\theta=-\pi}^{\pi}\int_{\tau=-D}^{D}\Im_\theta\left[\epsilon(k,\tau)\right]^2 d\theta d\tau = \\ &= \int_{\theta=-\pi}^{\pi}\int_{\tau=-D}^{D}|\Upsilon(\theta,\tau)|^2 d\theta d\tau = \\ &= \int_{\theta=-\pi}^{\pi}\int_{\tau=-D}^{D}|I_1(\theta)|^2\left|\left(e^{j\theta\tau}M(\theta)-H(\theta)-G(\theta)\tau\right)\right|^2 d\theta d\tau \\ &= \int_{\theta=-\pi}^{\pi}\int_{\tau=-D}^{D}|I_1(\theta)|^2\left|\Upsilon_0(\theta,\tau)\right|^2 d\theta d\tau,\end{aligned} \tag{18}$$

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

The error term $\Gamma(\mathbf{m},\mathbf{h},\mathbf{g})$ is constructed as a Weighted Least Squares (WLS) of the error $\Upsilon_0(\theta,\tau)$ 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 $|\Upsilon_0(\theta,\tau)|$ is required in frequencies where $|I_1(\theta)|$ is higher. However, this term also causes the energy of the resulting filters $M(\theta)$, $H(\theta)$ and $G(\theta)$ to be located in frequencies where $|I_1(\theta)|$ has low energy. This in turn means that the obtained filters are likely to be modulated, and applying them to the image may lead to an uninformative result, leading to inaccuracies. The correction of this behavior can be obtained by adding an additional term enforcing the designed filters to be located at frequencies where $|I_1(\theta)|$ is high. The overall penalty function is therefore

$$\begin{aligned}\Gamma_{\text{total}}(\mathbf{m},\mathbf{h},\mathbf{g}) &= \int_{\theta=-\pi}^{\pi}\int_{\tau=-D}^{D}|I_1(\theta)|^2\left|\Upsilon_0(\theta,\tau)\right|^2 d\theta d\tau + \\ &\quad +\alpha\int_{\theta=-\pi}^{\pi}\int_{\tau=-D}^{D}\left[1-|I_1(\theta)|^2\right]|\Upsilon_0(\theta,\tau)|^2 d\theta d\tau = \\ &= \int_{\theta=-\pi}^{\pi}\int_{\tau=-D}^{D}\left[\alpha+(1-\alpha)|I_1(\theta)|^2\right]|\Upsilon_0(\theta,\tau)|^2 d\theta d\tau.\end{aligned} \tag{19}$$

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 $0 \le \alpha \le \infty$. For $\alpha = 0$ we get the previous error term. For $\alpha = 1$ the error term contains a flat frequency weight.

The proposed criterion uses the knowledge on the maximal length of the motion vectors, and possibly, their distribution. Additionally, the specific image is considered in the minimization procedure as well. This is a natural and beneficial property since we expect that different images require different filters. From the application point of view, however, for a given pair of images, we have to compute their corresponding optimal filters, and then estimate their motion field. In cases

were we want to avoid these computations we can supply a set of filters which should match the spectral characteristics of typical images (such as $1/\theta$ as proposed by Simoncelli [8]).

The minimization term in equation (19) can be rewritten in a bilinear form $\Gamma_{\text{total}}(\mathbf{m},\mathbf{h},\mathbf{g}) = \mathbf{x}^H\mathbf{R}\mathbf{x}$, where the vector $\mathbf{x}$ contains all the filters coefficients, and the operation $\mathbf{x}^H$ is the complex conjugate of $\mathbf{x}$,

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

The matrix $\mathbf{R}$ is an $(3L+3)\times(3L+3)$ positive definite matrix, which depends on the interpolating function $b(x)$, the maximal motion D, and the spectral form of the image $I_1(k)$. Appendix B presents 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

$$\text{Minimize } \mathbf{x}^H\mathbf{R}\mathbf{x} \quad \text{Subject to } \quad \mathbf{x}^H\mathbf{x} = 1. \tag{21}$$

The solution of the above problem is the eigenvector of the matrix $\mathbf{R}$ corresponding to the smallest eigenvalue, and can be readily obtained using the SVD method [14].

3.3 The Relationship to Simoncelli's Filters

As mentioned above, Simoncelli also proposed a frequency domain WLS criteria for the design of the optimal filters. It can be shown that the filters that are suggested by Simoncelli are actually a particular case of the proposed approach, where

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

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

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

As can be seen, up to a constant we got the same optimality criteria as in Equation (8). Since Simoncelli's filters correspond to infitesimal motion, the estimation of larger motion vectors are expected to lead to larger errors.

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 - we found this choice to better suit the filters design), and that $b(x) = \text{sinc}(x)$ (the trivial choice). The question is: What are the optimal filters **m**, **h** and **g** in this case? Appendix C treats this question, showing 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**:

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

   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:

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

The implication of the above result is that optimal filters according to 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 specific filters design procedure.

4 Experiments and Results

In this Section we present several examples which demonstrate the ability of the proposed optimal filters to give better motion estimation performance. We start this section with the obtained filters. Figure 8 presents the 3 obtained 9-taps filters **m**, **h** and **g** for for 3 different values of D (the maximal 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 correspond to interpolated versions of the discrete filters using the sinc function.

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

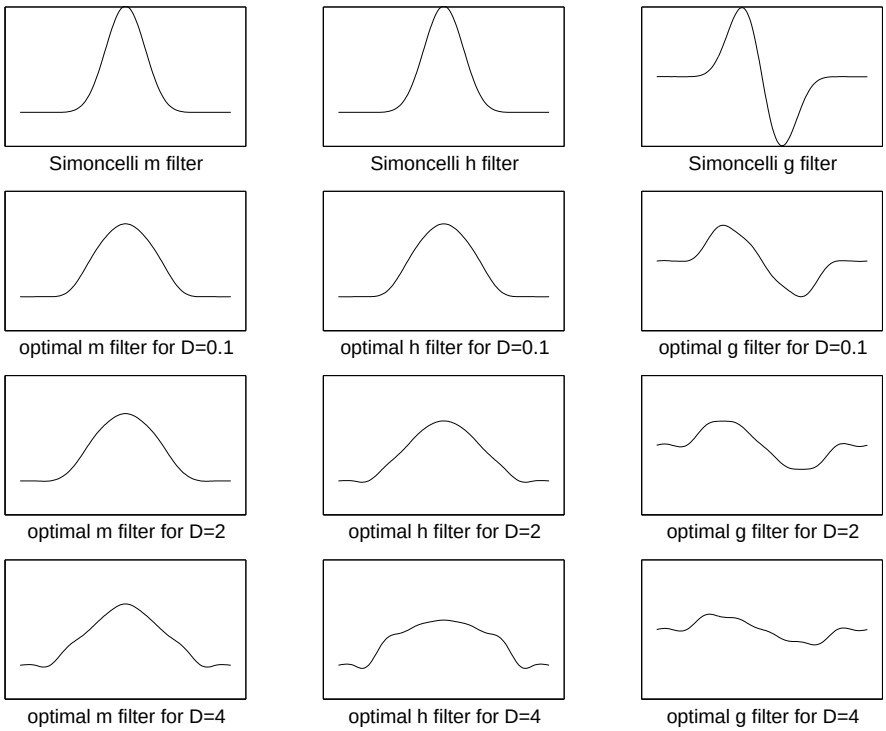


Figure 8: Simoncelli's filters (top row) and the optimal filters for $D = 0.1$ (second row), 2 (third row) and 4 pixels (last raw).

Recall that the proposed filters are the ones which obtain the minimum of the shiftable filters error averaged over all x and all τ in the range $[-D, D]$,

$$\int_x \int_{\tau=-D}^{D} \left(m(x+\tau) - h(x) - g(x)\tau\right)^2 d\tau dx.$$

Figure 9 presents a graph of this error as a function of τ (i.e., the integration over τ is removed in the error computation) for Barron 9-taps filters, Simoncelli's 9-taps filters, and the optimal 9-taps filters for $D = 2$ and $\alpha = 1$. As can be seen, Simoncelli and Barron filters results with exact 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, it is easy to notice that the overall error is much smaller with the optimal filters, because of moderate errors for large values of τ. The average error term (the value of the integral) for Barron, Simoncelli and the optimal filters is 0.172, 0.1782 and 0.0961 respectively. This experiment also reveals that Simoncelli's filters are comparable to Barron's - a fact that we found to be true in later experiments as well.

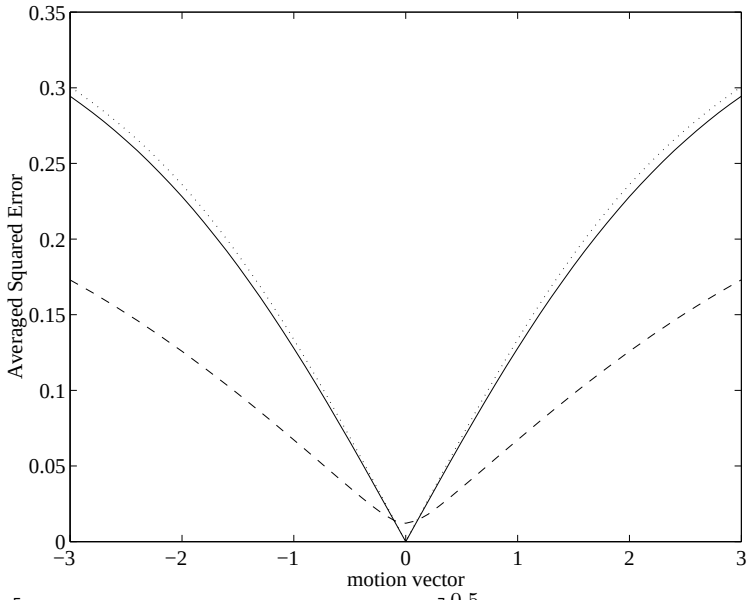


Figure 9: The error $\left[\int_x \left(m(x+\tau) - h(x) - g(x)\tau\right)^2 dx\right]^{0.5}$ as a function of τ for Barron (solid line), Simoncelli's (dotted line) and the optimal filters (dashed line).

Clearly, the above comparison does not prove that the obtained filters are indeed the optimal ones, since optimality (or at least better performance) should be proven with respect to optical flow estimation errors directly. This is the reasoning behind the next example. We take random pairs of vectors of length 160 having relative global translation between them in the range $\tau \in [-3, 3]$. We estimate the motion field using two different techniques: Lucas and Kanade [15], and Horn and Schunck [4] algorithms. We implement these two algorithms with three different sets of filters: Barron filters, Simoncelli's filters and the optimal filters.

Lucas and Kanade's algorithm suggests that for each pixel the BCCE of its local neighborhood should be combined to form an LS solution. We use uniform weights for 5 neighborhood pixels [15, 3]. Horn and Schunck's algorithm reconstructs the motion vectors by adding a smoothness penalty. We use the Laplacian operator for this penalty with $\beta = 300$. In order to reduce randomness effects, 10 such pairs are generated per each motion vector value. Figure 10 presents one of such ($\tau = 2$ pixels). Figure 11 presents the averaged absolute motion estimated error with Horn and Schunck, and Lucas and Kanade algorithms. Errors are presented for the Barron filters, Simoncelli's filters and the optimal filters.

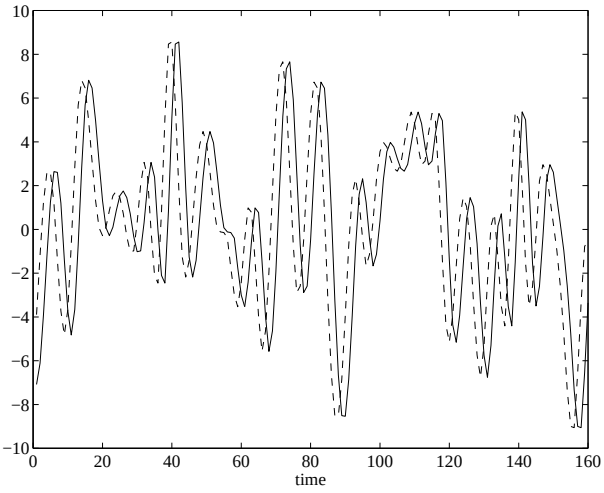
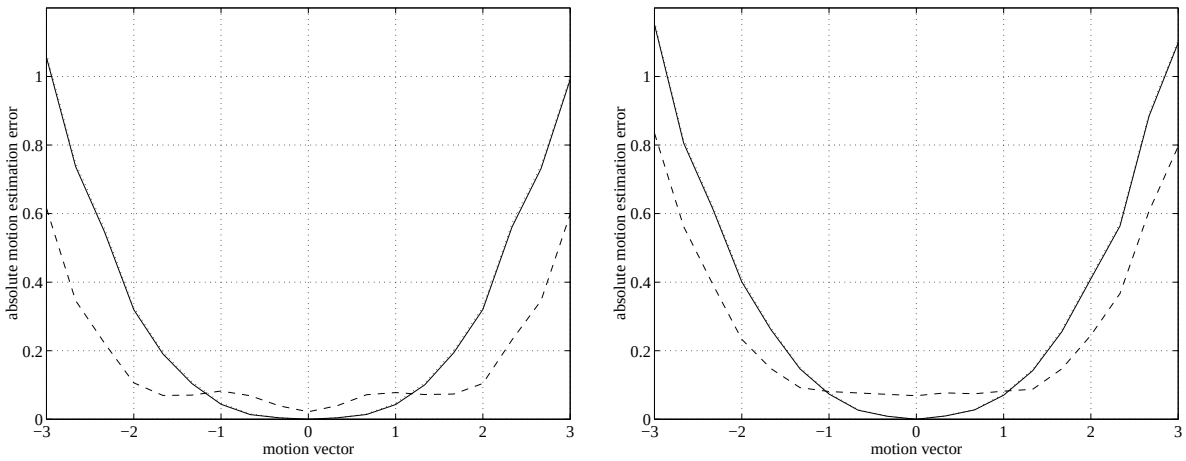


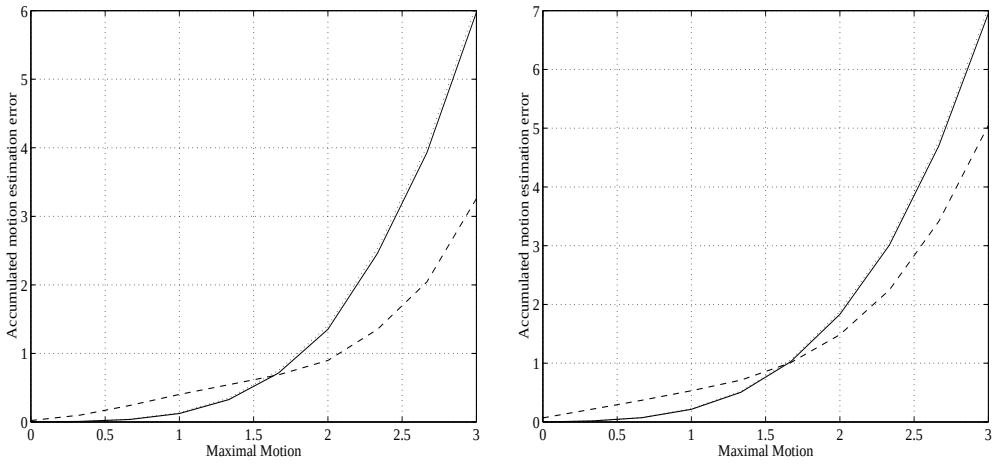
Figure 10: An example of a pair of vectors used for the presented tests.



(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)

The results show that the theoretic graph presented in Figure 9 indeed manifests the actual errors obtained in the motion estimation results. Figure 12 presents the accumulative error as a function of D in the range $D \in [0, 3]$, based on the results shown in Figure 11 (this graph presents the area beneath the error function in the range $\tau \in [-D, D]$ for different values of D).



(a) **(b)**

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)

We should note that in many optical flow estimation applications a multi-resolution approach is used to enable estimation of large motion vectors. In such cases, filters that lead to non-zero error near the origin are expected to perform poorer. The proposed methodology is general and can be used while forcing the error at the origin to be zero (either as a constraint or by a strong weight), thus getting a sub-optimal result with regard to the original penalty function, but one that leads to more practical filters when merged with multi-resolution approach.

Another issue 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) results sometimes (depends on the given image, number of taps, and the maximal motion value) in uninformative filters and instabilities. The following Figures describes the influence of α on the obtained filters. Figure 13 presents 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 reveal 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 mostly resolved. 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 good choice - it results in uniform frequency weight which no longer depends on the image spectrum. This in turn means

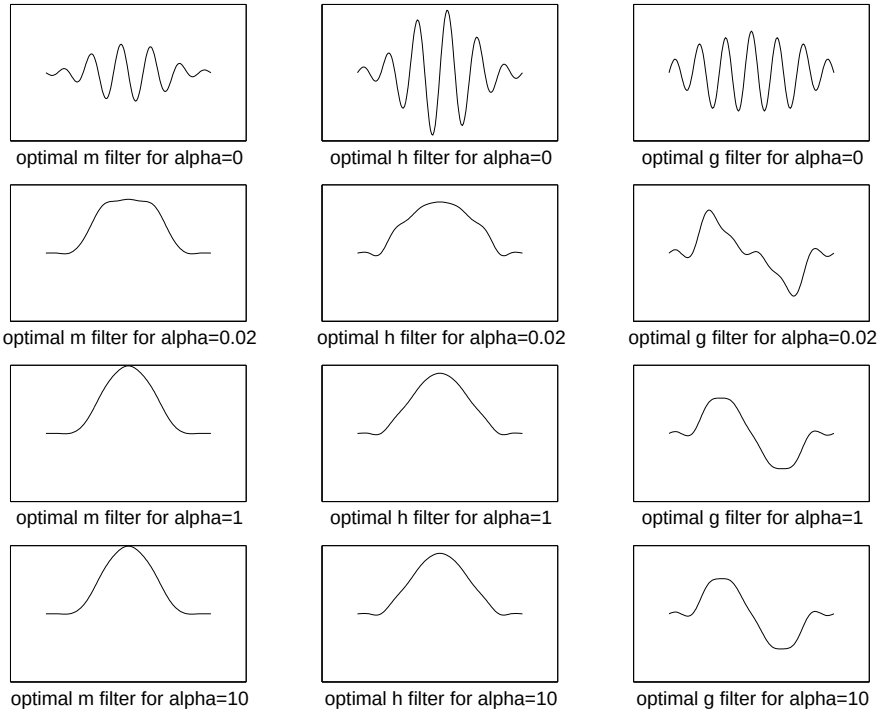


Figure 13: 9-taps optimal filters $\mathbf{m}$, $\mathbf{h}$ and $\mathbf{g}$ for $D = 2$ and $\alpha = 0$ (first row), 0.02 (second row), 1 (third row), and 10 (last row).

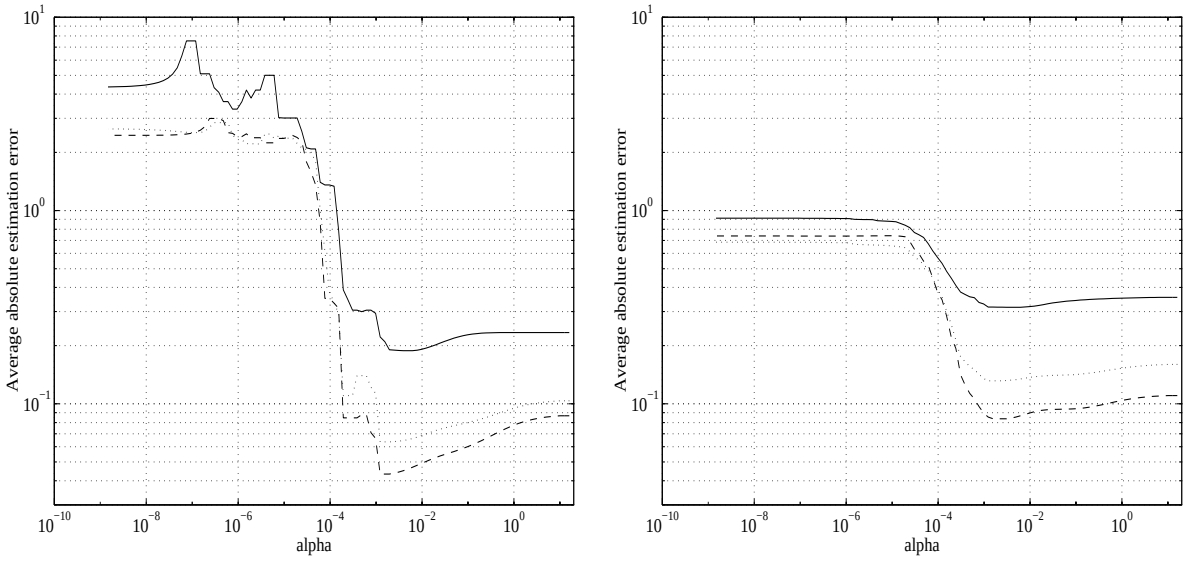
that this spectrum need not be evaluated. Beyond that, we see that the obtained filters are general and serve all the images, as they do not depend on the image characteristics.

Figure 14 presents the average absolute motion estimation error as a function of α for Horn and Schunck and Lucas and Kanade 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.

We can conclude from these 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. The 2-D case can be modelled 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 general BCCE relationship

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



(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

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

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

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 presents this composition of the filters.

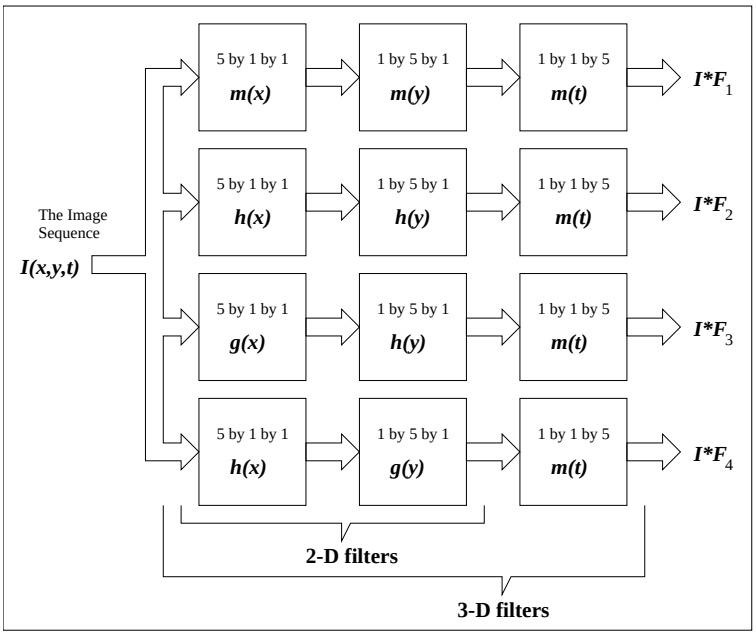


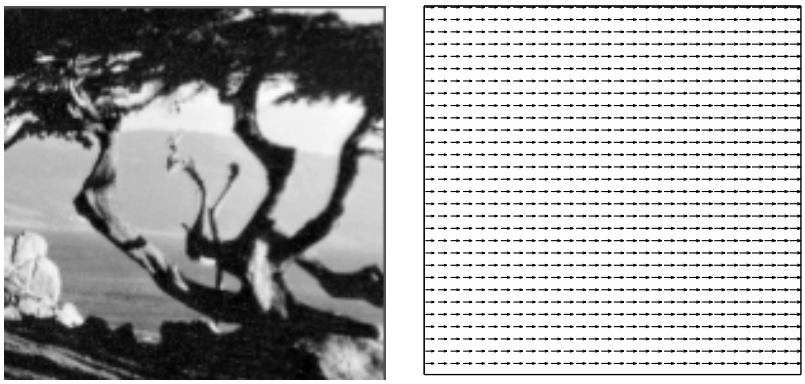
Figure 15: The proposed optimal filters configuration for the 2-D and 3-D cases.

Barron's filters were taken to be 11-taps pre-smoothing and 5 taps differentiator. 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 estimate the motion between three different pairs of images. We do not present the performance of the Simoncelli's filters since for this number of taps we got that Simoncelli's filters exhibit instability. Beyond that, based on the 1-D results we expect the performance of Simoncelli's filters to be similar to Barron's, once we use a different frequency weighting.

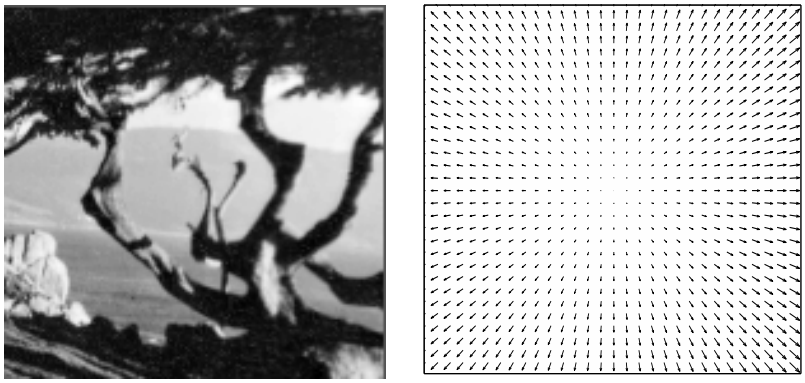
In all cases we construct 2-D filters from the given 1-D ones, based on the above composition. Figures 16, 17 and 18 show the three images we use, together with the true optical flow. These sequences, together with their true flow, are taken from Barron's WEB-site, and are called Translating Tree, Diverging Tree and Yosemite, respectively. In our simulations of motion estimation we use the Lucas and Kanade [15] algorithm with neighborhood of $[7 \times 7]$ pixels, weighted uniformly (adequate for smooth motion flow).

Figures 19 and 20 summarize the obtained results for the three sequences. Per each sequence we have computed the average angular error [3] for varying density values. We have also supplied the Mean Squared Error between the true and the estimated flow for varying density values. From the above results we can conclude several things:



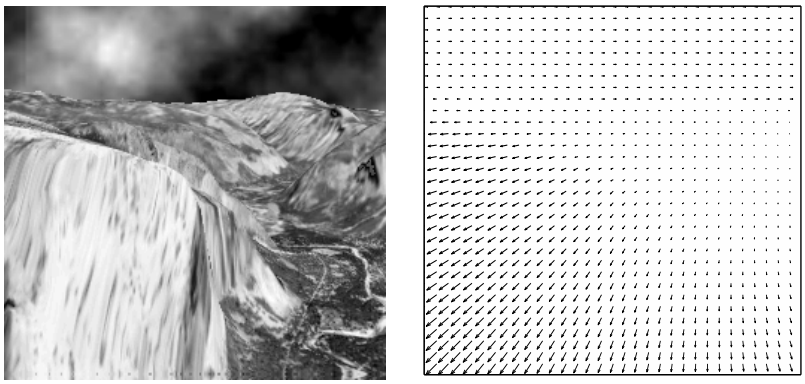
(a) **(b)**

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



(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

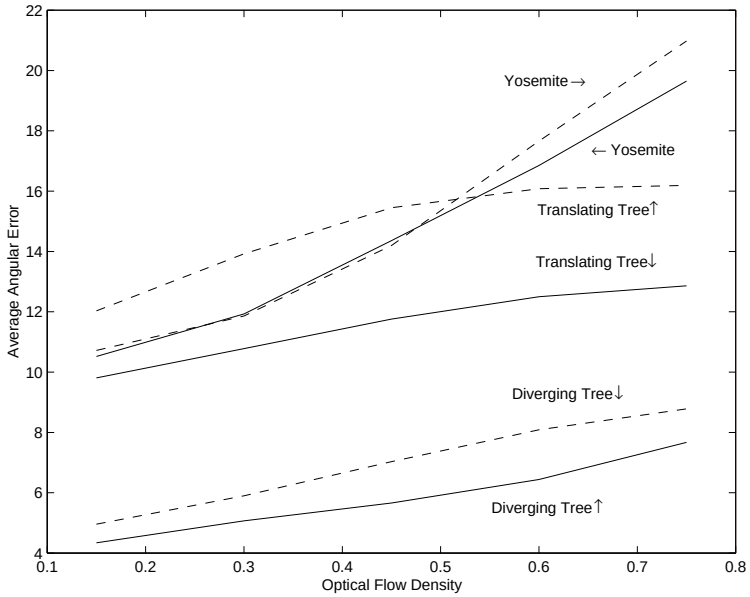


Figure 19: Average angular error of optical flow estimation for three image sets. For each image, the dotted line shows the deviation from the true optical flow using Barron's filters, while the solid lines using the proposed optimal filters.

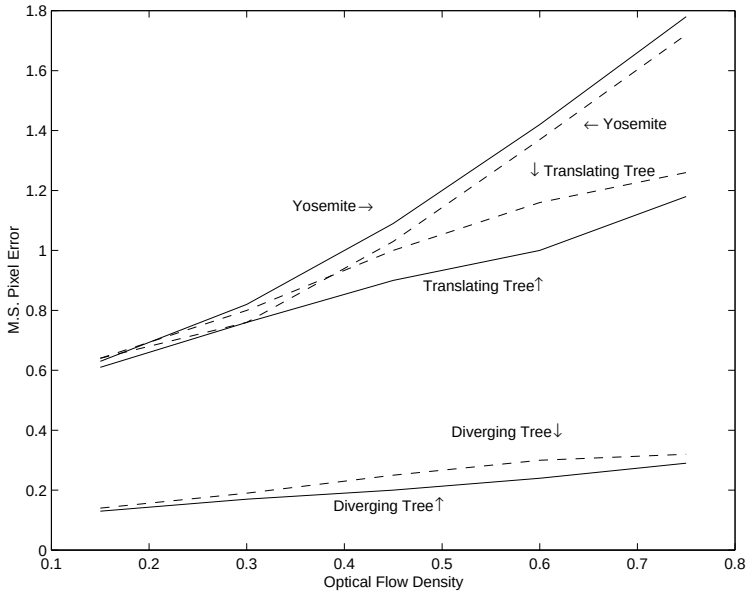


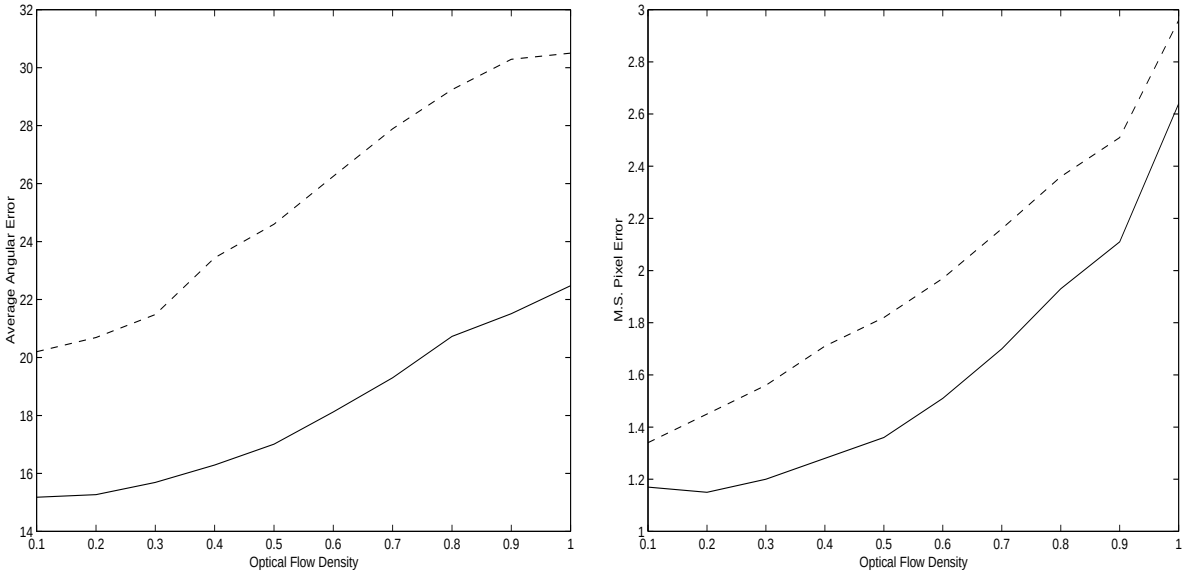
Figure 20: Average pixel error of optical flow estimation for three image sets. For each image, the dashed line shows the deviation from the true optical flow using Barron's filters, while the solid lines using the proposed optimal filters.

1. The confidence measure is the one proposed by Barron [3]. We can see that as the density grows 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 compute 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.

   Figure 21 shows that 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 better performance.



(a) **(b)**

Figure 21: Average angular error (left) and average pixel error (right), for the lower-left part (100×100) of the Yosemite image. In both graphs, solid lines plot the optimal filter results, and the dashed lines the Barron's filters results.

5 Summary and Conclusions

In this paper we have proposed a new design procedure for filters that 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 take into account the images spectrum, the transformation prior, and 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 benefit of using these filters. In our simulations we have studied the performance of Lucas and Kanade [15] (1-D and 2-D) and Horn and Schunck [4] (only for 1-D) algorithms. We have tested Barron's [3], Simoncelli's [8] and these new optimal filters.

This research could be extended in several ways: (i) 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; (ii) 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; (iii) 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 of 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; (iv) Instead of the Taylor expansion, we can use different expansion such as the Fourier series. This way we get the phase based motion estimation algorithms. The alternative BCCE in this case would be

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

(v) 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.

As a final remark, we should refer the interested reader to a recently introduced contribution by Robinson and Milanfar on the same topic [16, 17]. In their work, Robinson and Milanfar adopt an innovative statistical point of view to the problem, considering the Carmer-Rao lower bound

and the bias expression for a specific gradient-based motion estimation. They derive the best smoothing and derivative filters as those that minimize the overall motion estimation error bound. While there are some similarities between these two works, these are two quite independent and different methods addressing the same task.

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," *the Journal of Visual Comm. and Image Representation, Vol. 9, pp. 119-138, June*, 1998.

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

[9] M. Shin, D. Goldgof, K. Bowyer, and S. Nikiforou, "Comparison of edge detection algorithms using a structure from motion task," *IEEE Trans. Systems, Man, and Cybernetics - Part B: vol.31, No.4, August*, 2001.

[10] M. Elad, P. Teo, and Y. Hel-Or, "Optimal filters for gradient based motion estimation," in *International Conference on Computer Vision - ICCV 1999, Corfu, Greece*, pp. 559–565.

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

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

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

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

[15] 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.

[16] D. Robinson and P. Milanfar, "Fundamental performance limits in image registration," *IEEE Transactions on Image Processing, vol. 13, no. 9, pp. 1185-1199, September*, 2004.

[17] D. Robinson and P. Milanfar, "Bias minimizing filters for gradient-based motion estimation," in *Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA. November*, 2004.

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 true

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

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

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

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

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

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 Parseval's 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). □

B The Quadratic Form of the Optimization Problem

This appendix presents the detailed 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

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

the above error term can be rewritten as:

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

The notation $\otimes$ represents the Kronecker multiplication of matrices [14]. The internal integral, which corresponds to τ, can be computed analytically. The result is the expression

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

By defining the 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 $\Gamma_{\text{total}}(\mathbf{m}, \mathbf{h}, \mathbf{g}) = \mathbf{x}^H \mathbf{R} \mathbf{x}$. □

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)

$$\Gamma_{\text{total}}(\mathbf{m}, \mathbf{g}) = \int_{\theta=-\pi}^{\pi} \int_{\tau=-D}^{D} W(\theta) \left| \left(\left[e^{j\theta\tau} - 1\right] \hat{M}(\theta) - \hat{G}(\theta)\tau \right) \right|^2 d\theta d\tau \tag{B-9}$$

$$\Rightarrow \mathbf{R} \;=\; \int_{-\pi}^{\pi} W(\theta) \begin{pmatrix} 4D - 4\frac{\sin(\theta D)}{\theta} & \frac{2j[\theta D\cos(\theta D)+\sin(\theta D)]}{\theta^2} \\ \frac{2j[\theta D\cos(\theta D)-\sin(\theta D)]}{\theta^2} & \frac{2D^3}{3} \end{pmatrix} \otimes \mathbf{F}(\theta) d\theta.$$

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 $b(x) = \mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$. In this appendix we approximate the solution for the optimal 1-D filters $\mathbf{m}$, $\mathbf{h}$ and $\mathbf{g}$ and give a closed-form expression for their values. These filters 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 apply an approximation of the form

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

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 be 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 thus leads to the optimal solution. Since we have to zero these terms for each τ, minimum for this penalty is given by the conditions

1. $\forall\ k,\ \ \hat{m}(k) = \hat{h}(k)$,
2. $\forall\ k,\ \ \hat{g}(k) = \hat{m}'(k)$,
3. $\hat{m}(k)$ should be the non-trivial minimizer for 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 $\mathbf{m}$,

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

Regarding the third condition, we use the fact that

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

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

$$\begin{aligned} \Gamma(\mathbf{m}) = \sum_{k=-\infty}^{\infty} m''(k)^2 &= \sum_{k=-\infty}^{\infty} \left[\sum_j m(j) b''(k-j) \right]^2 = \\ &= \sum_{j_1} \sum_{j_2} m(j_1) m(j_2) \sum_{k=-\infty}^{\infty} b''(k-j_1) b''(k-j_2) \\ &= \sum_{j_1} \sum_{j_2} m(j_1) m(j_2) C(j_1 - j_2), \end{aligned} \tag{C-6}$$

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

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

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,

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

as described in sub-section 3.2.