ICCV 1999, Corfu, Greece, pp. 559-565, September 1999.

Optimal Filters For Gradient-Based Motion Estimation

Michael Elad
HP Laboratories-Israel
Technion City
Haifa 32000 Israel

Patrick Teo
Stanford University
Computer Science Department
Stanford CA, USA

Yacov Hel-Or
The Interdisciplinary Center
Computer Science Department
Herzliya, Israel

Abstract

Gradient based approaches for motion estimation (Optical-Flow) 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 spatio-temporal derivatives. 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; The suggested approach takes into account prior knowledge on the motion distribution, the image characteristics, and the allocated filter length. Simulations demonstrate the advantage of the new design approach.

1 Introduction

Estimating motion between two images plays a vital role in many applications and has drawn a lot of attention during the last two decades. There are many ways to approach this problem and indeed many algorithms have been proposed for this task, e.g. [6, 9, 1]. In Barron et. al. [1] 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 [6].

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. Denoting the intensity values of the image sequence by the function $I(x,y,t)$, where (x,y) is the spatial position and t is the temporal axis, the brightness constancy assumption along the image stream yields [6]:

$$\frac{dI(x,y,t)}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \qquad (1)$$

Defining $(u^x, u^y) = (\frac{dx}{dt}, \frac{dy}{dt})$, as the spatial velocity of each spatio-temporal point in the image sequence, we obtain

$$I_x u^x + I_y u^y + I_t = 0 \quad . \qquad (2)$$

Here I_x, I_y and I_t denote the spatial and temporal derivatives. This *Brightness Constancy Equation* (BCE), 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 constrains are required to uniquely recover the motion field. For this purpose, an assumption of smoothness (spatial [8, 1] and/or temporal [4, 2]) is typically imposed.

One issue that is critical to the implementation of the above BCE is that image derivatives are computed based on sampled information. It is commonly agreed [1, 11] that approximating the spatio-temporal derivatives by finite differences produces error in the above equation and subsequently in the estimated motion. One of the major conclusions of Barron [1] 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 applied to the image sequence prior to motion estimating [8, 1]. Since finite gradients are more accurate at low frequencies [8], pre-smoothing attenuates spatial and temporal aliasing effects, and improves the overall accuracy of gradient estimation.

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

$$\begin{aligned} &I_x u^x + I_y u^y + I_t = \\ &= \{F_1 * I\}u^x + \{F_2 * I\}u^y + \{F_3 * I\} = 0 \end{aligned} \qquad (3)$$

where F_1 F_2 and F_3 are spatio-temporal digital filters of some sort, and $\{A * B\}$ 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 [1, 11]. 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 demanded that F_1, F_2 and F_3 are separable [6, 1, 11]. We adopt this line of reasoning in this paper as well.

2 Existing Motion Estimation Filters

The numerical analysis literature contains many methods for approximating gradient filters [8]. Most of the papers describing optical flow estimation using the BCE apply simple gradient filters such as $\frac{1}{2}[-1,0,1]$ (e.g. [6]). In many papers the choice of these filters is not even mentioned.

In their original paper [6], 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 cube of $2 \times 2 \times 2$ pixels in the image sequence. These gradients refer to a center point of the cube (which means that the estimated flow corresponds to points between pixels). No motivation or justification for this choice of gradient estimation is given. According to Barron et. al. [1], these gradient filters are said to be a *"relatively crude form of numerical differentiation and can be the source of considerable error"*. Barron et. al. propose [1] the application of a $5\times5\times5$ 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 filter proposed by Barron is the 5-tap central-difference filter $\frac{1}{12}[-1,8,0,-8,1]$, which is the result of a design procedure described in [8]. In this scheme the goal is to obtain a near-accurate gradient transfer function $D(\theta) = j\theta$ where the filter coefficients $\{d(k)\}_{k=1}^{L}$ are designed to meet this requirement as closely as possible.

The derived $1D$ gradient filter is used to produce 3 types of derivatives (x-derivative, y-derivative, and t-derivative) in a separable manner: a 3-D $[5 \times 5 \times 5]$ pre-smoothing kernel is first applied to the image sequence. Then, each axis is differentiated separately applying the obtained derivative filter.

Figure 1 (upper graph) depicts the power spectrum of the gradient filter, $|\tilde{D}(\theta)| = |D(\theta)S(\theta)|$, compared to an analytic differentiation of the smoothing filter, $|j\theta S(\theta)|$. The pre-smoother is taken to be a sampled Gaussian filter with a variance of 1.5, and the gradient filter is the 5-taps central-difference filter as suggested by Barron et. al. It is demonstrated that the error between these two responses is very small for low frequencies but increases as the frequency tends to $\pm\pi$.

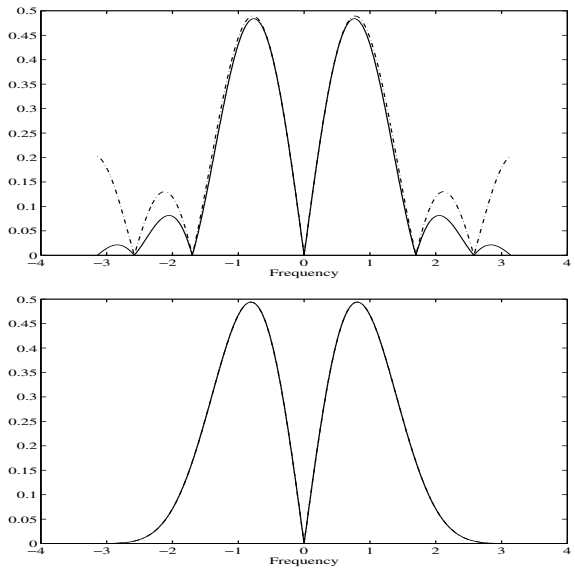

Figure 1: The derivative filters in the frequency domain as proposed by Barron (upper graph) and Simoncelli (lower graph). The solid line depicts the proposed filter and the dashed line depicts the analytic filter. Note that in the lower graph the analytic filter is actually coincides with the the proposed filter.

In [11] Simoncelli proposed that the pre-smoother and the derivative filters should be well-matched; that is, the filter $\tilde{d}(x)$ should be the first derivative of the filter $s(x)$. For digital filters, this requirement is stated with respect to some choice of interpolant.

If we denote a smoothing and a derivating 1-D filter pair in the frequency domain by $S(\theta)$ and $\tilde{D}(\theta)$ respectively, then the error $[j\theta\ S(\theta) - \tilde{D}(\theta)]$ can be minimized in a more accurate manner. For example, high frequencies which are not treated correctly by $\tilde{D}(\theta)$ can be attenuated by the pre-smoother $S(\theta)$ in order to minimize the above approximation error. Figure 1 (lower graph) shows the frequency response of the gradient filter, $\tilde{D}(\theta)$, compared to an analytic gradient of the smoother filter, $j\theta \cdot S(\theta)$. This time the error between these two is negligible.

2.1 Existing Approaches - Is It Really The Best We Can Do?

The existing methods for designing filters to be used in optical flow estimation, aim at obtaining filters which are as similar as possible to derivatives. However, all existing methods over-look the final goal of these filters, namely, the estimation of optical flow. In this paper we first propose a technique to derive a set of filters which are optimal specifically with respect to this goal. These filters are designed to give the best estimation results in term of accuracy, where their derivative characteristics are not a demand but a by-product. In our scheme we adopt useful design requirements from existing methods and add a few more:

1. First, and most important, minimization should be performed with respect to an energy function related to motion estimation error.
2. The motion characteristics should be considered. For example, assume that a-priori 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.
3. The design procedure should consider the number of taps allocated, and exploit them in the best possible manner.
4. The design should relate to an interpolated version of the discrete image sequence in order to refer to the continuous grid.
5. The design procedure should consider the characteristics of typical images, or better yet, the specific given image sequence.
6. If possible, the design should yield separable filters which are easier to implement.

While the first two requirements are novel to the proposed technique, the rest of these requirement are also considered by Simoncelli's method [11]. In the next section we present our approach which takes into consideration all the above requirements.

3 The Proposed Approach: 1-D Case

In this section we first develop the general case of transformation between two 1D signals $I_1(x)$ and $I_2(x)$. The extension to 2D images $I(x, y)$ is straightforward. In the general case, the transformation between these two signals can be of any transformation group $T(\mathbf{a})$, where $\mathbf{a}$ is a parameter vector defining the amount of transformation, i.e. $I_2(x) = I_1(T(\mathbf{a})x)$. In the simplest case, one signal is translated with respect to the other: $T(a)x = x + a$, where a is the amount of translation between the signals. In a different case, $T(a)x = e^a x$, and I_2 is a scaled version of I_1. Note, that where $a = 0$, $T(a)$ is the identity operator and $T(0)x = x$. More complicated transformations can be multi-parameter groups, for example: $T(\mathbf{a})x = e^{a_1}x + a_2$, which is a composition of scaling (by a_1) and translation (by a_2). In all these cases, our goal is to estimate the transformation parameters $\mathbf{a}$.

Assuming $\mathbf{a}$ is small, the first order Taylor expansion of I_2 around $\mathbf{a} = \mathbf{0}$ yields:

$$I_2(x) = I_1(T(\mathbf{a})x) = I_1(x) + \frac{\partial I_1}{\partial x}(x)\left(\frac{\partial x}{\partial \mathbf{a}} \cdot \mathbf{a}\right) \quad (4)$$

where the dot defines an inner product, and the derivative with respect to $\mathbf{a}$ is $\frac{\partial x}{\partial \mathbf{a}} = \left[\frac{\partial x}{\partial a_1}, \frac{\partial x}{\partial a_2}, \cdots\right]$ which is performed around $\mathbf{a} = 0$. The term $I_1(x)$ in the right-hand side is due to the fact that $I_1(T(0)x) = I_1(x)$.

For example, in a motion model of pure translation, $T(a)x = x + a$, and Equation 4 becomes $I_2(x) = I_1(x) + a\frac{\partial I_1}{\partial x}$. In the case of scaling, $T(a)x = e^a x$, and we get $I_2(x) = I_1(x) + ax\frac{\partial I_1}{\partial x}$. In a multi-parameter transformation we get more terms in the expansion.

Note, that Equation 4 is defined over the continuous domain. In practice, however, we obtain a sampled version of the continuous signal, $I_1(x_k)$ and $I_2(x_k)$, where $k = 1 \cdots L_I$. Using an interpolating function $b(x)$ we can approximate the continuous signals $\hat{I}_1(x)$ and $\hat{I}_2(x)$ by:

$$\begin{aligned} \hat{I}_1(x) &= \sum_{k=1}^{L_I} I_1(x_k) b(x - x_k) = b(x) * I_1(x_k) \quad (5) \\ \hat{I}_2(x) &= \sum_{k=1}^{L_I} I_2(k) b(x - x_k) = b(x) * I_2(x_k) \end{aligned}$$

The choice of $b(x)$ can be, for example, the *sinc* function or a more "gentle" function such as the Gaussian.

Substituting the continuous approximation into Equation 4 we get:

$$b(x) * I_2(x_k) \approx b(x) * I_1(x_k) + \frac{\partial b(x)}{\partial x} * I_1(x_k)\frac{\partial x}{\partial \mathbf{a}} \cdot \mathbf{a}$$

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 former. We relaxed this choice and permit three (might be different) filters to estimate the optical flow.

In this manner, the optimal filters are designed with maximal flexibility. In this general form, we are looking for three filters $m(x)$, $h(x)$, and $g(x)$, such that:

$$m(x)*I_2(x_k) \approx h(x)*I_1(x_k)+g(x)*I_1(x_k)\frac{\partial x}{\partial \mathbf{a}}\cdot\mathbf{a} \quad (6)$$

According to the above assumption $m(x)*I_2(x_k) = m(x) * I_1(T(\mathbf{a})x_k)$. This in turn, means that applying the filter $m(x)$ to the signal I_1 can be performed equivalently by first applying the (inverse) transformation to the filter, and then convolving with the original signal I_1, that is: $m(x) * I_1(T(\mathbf{a})x_k) = m(T^{-1}(\mathbf{a})x) * I_1(x_k) = m(x) * I_2(x_k)$, where $T^{-1}(\mathbf{a})$ is the inverse transformation of $T(\mathbf{a})$. For a proof of this step, the reader is referred to [5]. Now, we can eliminate reference to I_2, and rewrite equation 6:

$$\left(m\left(T^{-1}(\mathbf{a})x\right) - h(x) - g(x)\frac{\partial x}{\partial \mathbf{a}}\cdot\mathbf{a}\right) * I_1(x_k) = \\ = \varepsilon(x,\mathbf{a}) \approx 0$$

The optimality of the filters is designed with respect to this equation. The filters, m, g, and h, are optimal if they minimize the expected value of $\varepsilon^2(x,\mathbf{a})$, where the expectation is performed over all actual x and $\mathbf{a}$:

$$\Gamma(m,h,g) = \int_x\int_{\mathbf{a}} \varepsilon^2(x,a)dxd\mathbf{a} \quad (7)$$

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, $w(a)$, into the internal integration, requiring smaller errors for the more probable motion vectors.

Unlike the minimization problem which computes derivative filters, the signal I_1 is included in the minimization. This allows one to design optimal filters over a collection of signals or more practically, over signals bearing certain properties, for example, a decaying power spectrum with a certain exponential decay constant (such as in natural images). Alternatively, we can remove the dependence over the choice of signals by computing the error only with respect to the filter term: $\|m(T^{-1}(\mathbf{a})x) - h(x) - g(x)\frac{\partial x}{\partial \mathbf{a}}\cdot\mathbf{a}\|$.

In the next section we elaborate the derivation of optimal filters for a translational motion model. Other transformation groups can be handled in a similar manner and will be not presented here.

3.1 Optimal Filters for 1D Translation

In the translational motion model we assume that $I_2(x) = I_1(x+a)$ where a is the amount of translation. Note that this model assumes a translated motion only *locally* (in the range of the filter support). The global motion does not have to be a pure translation. However, this approximation is valid only for smooth motion fields. In this case, $T^{-1}(a)x = x - a$ and Equation 7 becomes:

$$\varepsilon(x,a) = (m\,(x-a) - h(x) - ag(x)) * I_1(x_k)$$

In the above equation, the function $m(x)$, shifted by a, is approximated by $h(x) + ag(x)$. This resembles the concept of **"Shiftable Filters"** as defined in [10, 5], where a transformed version of a function is expressed as a linear sum of a set of basis functions.

Assuming that a is bounded within the range $|a| \leq D$, the optimal filters are those that minimize the error term:

$$\Gamma(m,h,g) = \int_x\int_{a=-D}^{D} \varepsilon(x,a)^2 dxda \quad (8)$$

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

$$\Gamma(m,h,g) = \int_\theta\int_a \Im_\theta\left[\epsilon(x,a)\right]^2 d\theta da = \quad (9)$$

$$= \int_\theta\int_a |I_1(\theta)|^2 \left|\left(e^{j\theta a}M(\theta) - H(\theta) - G(\theta)a\right)\right|^2 d\theta da$$

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, h, g, and I_1, respectively. The integration of θ is performed from $-\pi$ to π and the integration of a from $-D$ to D. In the above energy functional, the filters are specified as continuous functions; in practice, the optimal set of *digital* filters satisfying this equation is sought instead, e.g. $m(k), h(k)$ and $g(k)$ for $k = -L\cdots L$. As a result, some suitable interpolant, $b(x)$, is assumed, where:

$$\hat{m}(x) = \sum_{k=-L}^{L} m(k)b(x-k)$$

and similarly for $\hat{h}(x)$ and $\hat{g}(x)$. If we take the *sinc* function as our interpolant, we obtain:

$$M(\theta) = \sum_{k=-L}^{L} m(k)exp\{-jk\theta\}$$

and similarly for the other filters.

Arranging all the unknowns of the filters in a vectorial form:

$$\mathbf{x} = [m(-L)\ldots m(L), h(-L)\ldots h(L), g(-L)\ldots g(L)]^T$$

it is possible to rewrite $\Gamma(m,h,g)$ in a bilinear form

$$\Gamma(m,h,g) = \mathbf{x}^T\mathbf{R}\mathbf{x} \quad ,$$

where the matrix $\mathbf{R}$ is 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 $I_1(k)$. For more information about the content of this matrix the interested reader is referred to [3] . The optimal filters $\mathbf{x}$ are calculated solving the following minimization problem:

$$\mathbf{x} = \arg \min_{\|\mathbf{x}\|=1} \Gamma(m, h, g)$$

Minimization over $\|\mathbf{x}\| = 1$ is introduced to avoid the trivial solution $\mathbf{x} = 0$. 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 [7].

4 Relationship to Derivative Filters

Two important features distinguish the proposed method from other methods of designing optimal derivative filters. First, by allowing the filter m to differ from the filter h (that is, the pre-smoothing filters applied to the two images may be different), the estimated motion when there is no motion could be small but non-zero (since $m - h \neq 0$). However, this relaxation results in a reduction in error for the large non-zero translations, and the filters are optimal in a least-squares sense. Second, and more important, the characteristics of the expected motion can be explicitly specified. Thus, the motion estimation filters can be designed to be optimal with respect to a particular class of motion. The traditional derivative filters are actually a special case of those designed by our approach; It can be shown that when $D \to 0$, i.e. when the expected motion tends to zero, our approach yields the optimal derivative filters as suggested by Simoncelli [11]. For more details the reader is referred to [3].

5 Results

In this section we present several examples that demonstrate the ability of the proposed optimal filters to give better motion estimation performance. Figure 2 shows the three filters (9-taps) m, h and 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 were obtained with $|I_1(\theta)| = 1$ (which means that no frequency weight is involved). For comparison, Simoncelli's 9-taps derivative filters ($m = h$) are given as well. All these graphs plot interpolated versions of the discrete filters (using *Sinc* interpolations). As expected, when D is very small we get $m = h$. Another property that can be seen from these graphs is the better exploitation of the filters support.

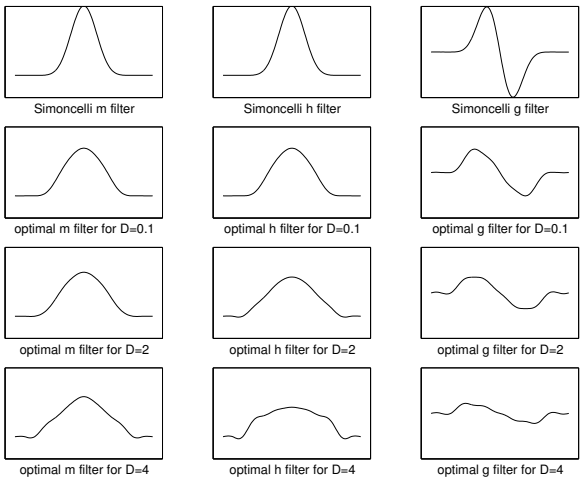

Figure 2: Simoncelli's filters (top row) and the optimal filters for $D = 0.1$ (second row), 2 (third row) and 4 pixels (last raw). The filters m, h, and g, are plotted in the left, middle, and right column, respectively.

As D increases, the obtained filters become wider (for the same number of taps). This seems intuitive since, with high speeds, aliasing will occur at all but the lowest spatial frequencies, so smoothing should be applied in a more drastic manner.

Recall that the proposed filters are the ones which minimize the error $(m(x-a) - h(x) - g(x)a)^2$, averaged over all x and all a in the range $[-D, D]$. Figure 3 shows a graph of this error as a function of a for three sets of filters: Barron's 9-taps filters, Simoncelli's 9-taps filters, and our optimal 9-taps filters for $D = 2$. As can be seen, Simoncelli's and Barron's filters have zero error for $a = 0$, whereas the optimal filters give non-zero error. This comes from the fact that in the optimal case $m \neq h$. Beyond that, notice that the overall error is much smaller with the optimal filters (because of moderate errors for large values of a). 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.

We also tested the performance of the proposed filters on real images. We compared the results of applying the proposed filters with the results of applying Barron's filters. We have not simulated the performance of the Simoncelli's filters, since, based on the 1-D results, we expect the performance of these filters to be similar to Barron's.

In 2D signals (images) subject to 2D translation,

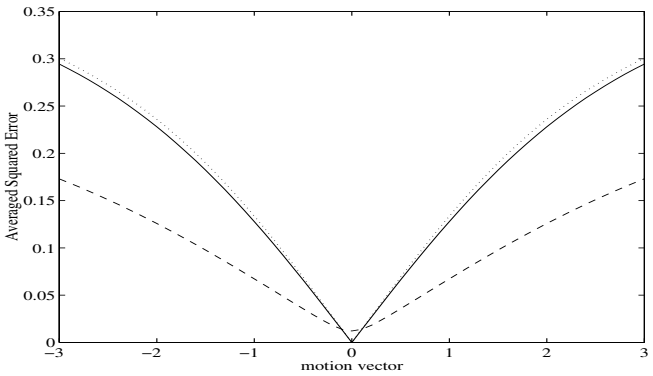

Figure 3: The error $\varepsilon^2(x,a)$ as a function of a for Barron (solid line), Simoncelli's (dotted line) and the optimal filters (dashed line).

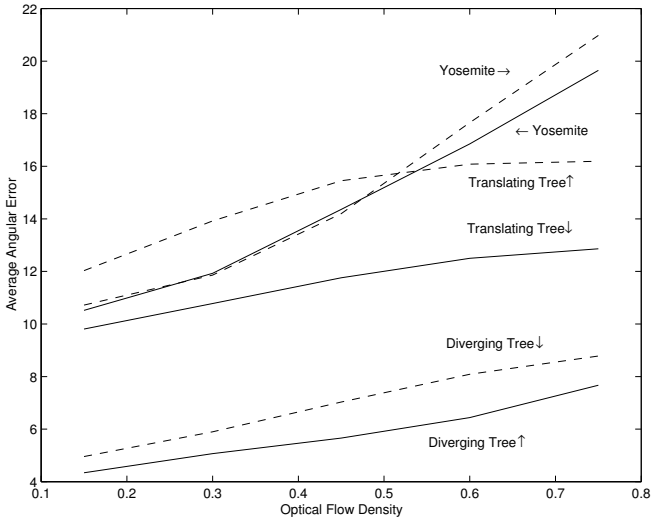

Figure 4: Average angular error of optical flow estimation of three images. For each image, the dotted line shows the deviation from the true optical flow using Barron's filters, while the solid lines using the proposed optimal filters.

Equation 6 becomes:

$$m_{2D} * I_2(x_k, y_j) \approx (h_{2D} + a_x g_{2D} + a_y f_{2D}) * I_1(x_k, y_j)$$

We derived these filters using 1D optimal filters, in a separable manner:

$$m_{2D} = m(x) * m(y) \quad ; \quad h_{2D} = h(x) * h(y)$$
$$g_{2D} = g(x) * h(y) \quad ; \quad f_{2D} = h(x) * g(y)$$

where m, g, and h, were derived as above.

Barron's filters were taken to be 11-taps presmoothing and 5 taps gradient filter. In order to apply an objective comparison, we used $[11 \times 11]$ taps optimal filters. These filters were designed as 1-D 11-taps filters with $D = 2$.

We tested the results on three images that were taken, together with the true optical flow from Barron's WEB-site, and are called Translating Tree, Diverging Tree and Yosemite, respectively.

In our simulations we estimated the motion using Lucas and Kanade's [9] algorithm with a neighborhood of $[7 \times 7]$ pixels, weighted uniformly (adequate for smooth motion flow).

Figures 4 and 5 summarize the obtained results for the three sequences. Per each sequence we have computed the average angular error [1] 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 it can be seen that the optimal filters yields, for almost all cases, better estimation results than those given by Barron's filters. This is certainly true for the first and the second sequences. The results are comparable for the Yosemite sequence. This 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 vectors, for which Barron's filters are nearly optimal. In order to better understand this behavior, we 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. Figure 6 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.

6 Conclusions

In this paper we 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 related to the motion estimation error. The design procedure can take into account the image spectrum, the transformation prior, and the available number of taps.

In the context of the proposed optimal filters, there are several issues that can be further considered:

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

$$m(x+a) = h(x) + g(x) \cdot a + f(x) \cdot a^2$$

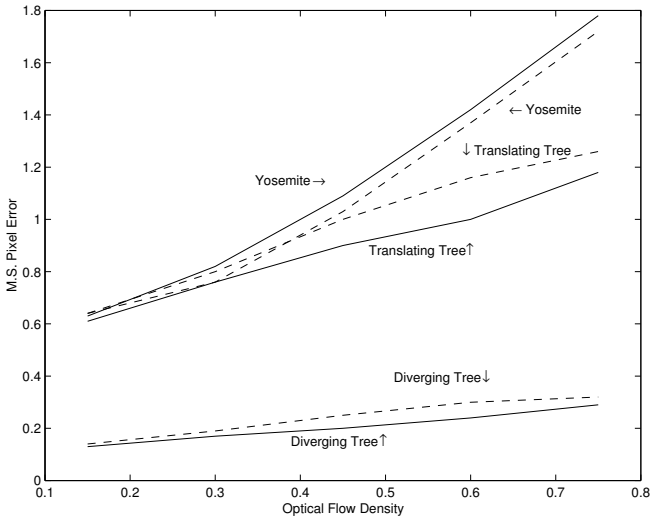

Figure 5: Average pixel error of optical flow estimation of three images. For each image, the dashed line shows the deviation from the true optical flow using Barron's filters, while the solid lines using the proposed optimal filters.

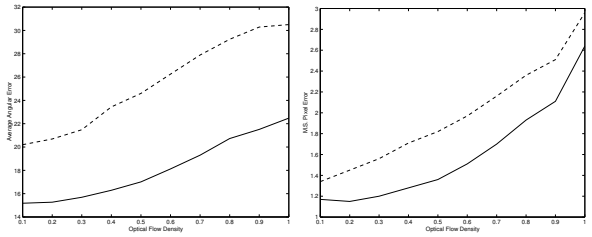

Figure 6: Average angular error (left) and average pixel error (right), for the lower-left part (100×100) of the Yosemite image. In both graphs, solid lines plot the optimal filter results, and the dashed lines the Barron's filters results.

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.

- Instead of the Taylor expansion, we can use different expansions such as the Fourier series. With this expansion we obtain the phase based motion estimation algorithms. The alternative BCE in this case would be, for example:

$$m(x+a) = h(x) + g(x) \cdot \sin(a) + f(x) \cdot \cos(a)$$

The different expansion can be advantageous in cases where it spans the motion field more precisely. In the case of Fourier expansion, it is also possible to exploit the shift invariant property in cases where recognition is required rather than estimation of the motion parameters.

References

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

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

[3] M. Elad, P. Teo, and Y. Hel-Or. Optimal filters for gradient-based motion estimation. Technical Report TR-97-111, HP Lab, 1997.

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

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

[6] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Inteligence*, 17:185–203, 1981.

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

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

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

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

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