BIAS-MINIMIZING FILTERS FOR MOTION ESTIMATION

Dirk Robinson and Peyman Milanfar

Electrical Engineering Department, University of California Santa Cruz
Email: {dirkr, milanfar}@ee.ucsc.edu

ABSTRACT

Among the myriad of techniques used in estimating motion vector fields, perhaps the most popular and accurate methods are the so called *gradient-based* methods. A critical step in the gradient-based estimation process is the estimation of image gradients using derivative filters. It is well known that the gradient-based estimators contain significant deterministic bias relating the gradient calculation. In this paper, we describe the fundamental relationship between estimator bias and derivative filters. From this, we suggest an image adaptive method addressing the design of bias-minimizing gradient filters. Simulations validate the superior performance of such filters for the many variants of gradient-based estimation including the widely used multiscale iterative methods.

I. INTRODUCTION

Among the myriad of techniques used in estimating motion vector fields, perhaps the most popular and accurate methods are the so called *gradient-based* methods [1]. These methods estimate image motion by relating the change in image intensity between images to spatial image gradients. These methods rely on the intensity conservation assumption that the underlying model for image dynamics follows the form

$$z_1(m,n) = f(m,n) + \epsilon_1(m,n) \tag{1}$$

$$z_2(m,n) = f(m - v_1(m,n), n - v_2(m,n)) + \epsilon_2(m,n) \tag{2}$$

where $\epsilon_i(m,n)$ is typically modelled as white Gaussian noise with variance σ^2. We refer to the indices m,n for the M by N image as the sample indices for the sampled functions $f(mT,nT)$ where $f(x,y)$ represents the underlying continuous image. The estimation of the motion vector field is commonly referred to as motion estimation or optical flow estimation.

As the name implies, the gradient-based estimation methods require measurement of image gradient. These measurements almost always take the form of simple, linear phase, linear shift invariant filters. Even though these filters play a vital role in the estimation scheme, and in fact have been shown to produce biased motion estimates [2], [3], [4] relatively few researchers have studied the design of such filters for optical flow estimation [5], [6]. Rather than study the errors introduced by such gradient filters, many researchers treat these errors as random in nature and attempt to design robust estimation methods to minimize the effects of such errors.

In this paper, using the bias formulation presented in [4], we derive bias-minimizing gradient filters in a fashion similar to [6]. We set up an optimization scheme whereby a bias-minimizing gradient filter is designed based on the image under observation. We experimentally verify the bias-minimizing property of such filters in both a non-iterative and multiscale iterative framework. Finally, we conclude by suggesting future work.

II. BIAS IN GRADIENT-BASED ESTIMATORS

To motivate our filter design methodology, we first explore the gradient-based motion estimation technique and the bias inherent to such methods. To simplify the analysis, we initially present the ideas using the 1-dimensional analogue of motion estimation. For the 1-D case, we suppose that the measured data is of the form

$$z_1(k) = f(k) + \epsilon_1(k) \tag{3}$$

$$z_2(k) = f(k - v(k)) + \epsilon_2(k) \tag{4}$$

THIS WORK WAS SUPPORTED IN PART BY THE NSF GRANT CCR-9984246.

In the derivation of the gradient-based estimator, the data is reformulated as $z(k) = z_1(k) - z_2(k) = f(k + v(k)) - f(k) + \epsilon(k)$ where ϵ is a Gaussian white noise random field with variance σ^2 and $v(k)$ is the unknown motion vector field.

Most of the gradient-based methods simplify the estimation problem by assuming that the unknown vector field $v(k)$ is comprised of locally parametric vector fields. The simplest of all models is the translational model of motion where the vector field is assumed to be constant over some region in space $v(k) = v,\ \forall k \in \Omega_i$ where Ω_i is a local region in the image space or the entire image itself. The gradient-based method generates an overdetermined set of linear equation by linearizing the function $f(k+v)$ about a point $v = 0$ in a Taylor series. The expansion looks like $f(k+v) - f(k) = vf'(k) + R(k,v)$ where the remainder term in the Taylor expansion R is ignored to produce a linear estimator for the velocity v,

$$\hat{v} = \frac{\sum_{k\in\Omega_i} f'(k)z(k)}{\sum_{k\in\Omega_i} (f'(k))^2} \tag{5}$$

In practice, the gradients (derivatives) were approximated using a gradient filter $g(k)$ applied to one of the available image $\tilde{f}'(k) = g(k) * z_1(k)$ (where $*$ represents convolution).

It has been noted on numerous occasions in the past [2], [3], [4] that the gradient-based estimators produce biased estimates. In [4], it was shown that the estimator bias

$$b(v) = E\left[\frac{\sum_{k\in\Omega_i} \tilde{f}'(k)z(k)}{\sum_{k\in\Omega_i} (\tilde{f}'(k))^2}\right] - v \tag{6}$$

is dominated by deterministic modelling error for high SNR situations. For many computer vision applications, the effective SNR falls into this high SNR regime. This deterministic bias results from a combination of the linearized data model and the gradient approximation. The approximate function for the deterministic bias in [4] is given in the frequency domain as

$$b(v) = \frac{\int_{-\pi}^{\pi} |F(\theta)|^2 \left[G(\theta)\sin(v\theta) - vG^2(\theta)\right] d\theta}{\int_{-\pi}^{\pi} |G(\theta)F(\theta)|^2 d\theta}. \tag{7}$$

where $F(\theta)$ is the Fourier transform of the image function (we assume that the image is sampled above the Nyquist rate) and $G(\theta)$ is the transform of the gradient filter. It is this bias function that we use to set up a process for designing gradient filters.

III. DESIGNING BIAS-MINIMIZING GRADIENT FILTERS

The design of gradient filters for use in motion estimation has been addressed previously in [5], [6]. In [5], it was noted that the gradient filters should be designed to match the actual derivative to the continuous function reconstructed using a specific interpolation kernel (assumed to be a Gaussian kernel). The filters of [5] have been noted to improve estimation performance [3], though the filters were not designed specifically for the purpose of motion estimation. In [6], the ideas were extended to address the specific problem of gradient-based motion estimation by designing a set of pre-smoothing and gradient filters to minimize modelling error. Unfortunately, this method minimizing the modelling error fails to address the interaction of these modelling errors with the structure of the estimator. Intuitively, in [6] the energy in the modelling errors is minimized over a range of unknown translations. The authors note that minimizing the error alone will not provide good filters since the optimization tends to create filters which contain most of their spectral energy at frequencies where the image spectral energy is lowest. They correct this by adding a Lagrangian penalty to focus the filters on high energy spectral regions. In our work, we minimize the energy in the estimator bias due to such modelling errors. This acts as the ideal penalty function by taking into account the structure of the estimator. Furthermore, our optimization process provides one optimal gradient filter which minimizes estimator bias instead finding three separate filters as in [6].

From (7) we see that the bias depends on three factors: the image content f, the gradient filter g, and the unknown translation v. We start from the assumption that translation is limited to some range $v \in [-V, V]$. From this we construct the following cost function for a particular image

$$\begin{aligned} J(\mathbf{a}) &= \int_{-V}^{V} b^2(v, F(\theta), \mathbf{a}) dv \\ &\approx \sum_{v_i} b^2\left(v_i, F(\theta), \mathbf{a}\right) \end{aligned} \tag{8}$$

where $\mathbf{a}$ is the vector of filter coefficients such that $G(\theta) = 2\sum_i a_i \sin(\theta i)$. Because the cost function is highly nonlinear in the unknown variables, we rely on a simple black box optimization routine `fminunc` provided by Matlab. Such an filter design method provides bias minimizing gradient filters for a given image. The optimization process can be used to find any linear phase antisymmetric derivative filter with $2N+1$ taps by optimizing over the N filter coefficients $\mathbf{a}$.

As an example, we construct a bandlimited signal $f(k) = \sum_{d=1}^{D} \frac{1}{d} \sin(\frac{\pi k d}{100} - \phi_d), k = 1 \ldots 100$ where ϕ_d is a fixed phase generated by drawing from a uniform distribution. We performed an optimization to find a 5 tap filter to compare with three popular gradient filters from [1], [7] which we refer to as the Fleet, Nestares, and Central (for central difference kernel) filters. The bias-minimizing filters were designed assuming a translation range of $v \in [-2, 2]$. We then computed actual estimator bias by applying each filter pairs of signals perfectly shifted by construction. The resulting biases are shown in Figure 1. When comparing the various filters, it becomes clear

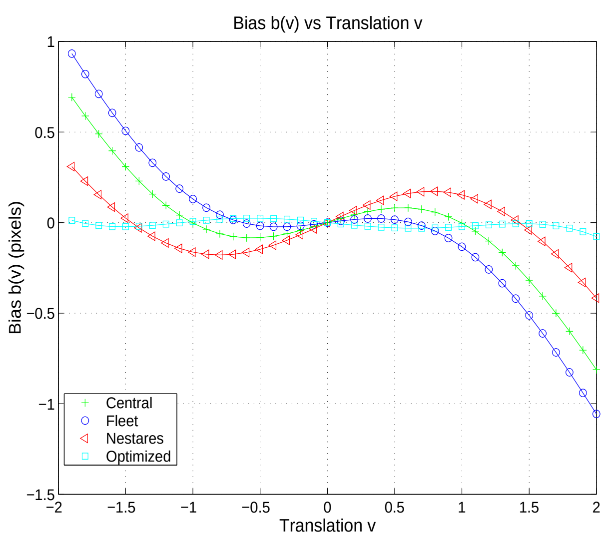

Fig. 1. Estimator bias $b(v)$ vs translation v for different gradient filters

that the optimized filter minimizes the estimator bias within this region. In fact, we found that the bias for the optimized filter outperformed the other filters outside the region of optimization as well.

IV. 2-D MULTISCALE ESTIMATION

While the computation of cost function is simple enough for the 1-D version of the problem, we found that further simplifications were necessary to reduce the complexity of the costly integration required in multidimensional case. To address this concern we make the following approximations. Using vector notation, $\mathbf{v} = [v_x, v_y]^T$ and $\underline{\theta} = [\theta_x, \theta_y]^T$ and $\mathbf{G}(\underline{\theta}) = [G_x(\underline{\theta}), G_y(\underline{\theta})]^T$. we can express the 2-D gradient-based estimator bias as

$$\mathbf{b}(\mathbf{v}) = \mathbf{Q}^{-1} \int |F(\underline{\theta})|^2 \mathbf{G}(\underline{\theta}) \sin(\underline{\theta}^T \mathbf{v}) d\underline{\theta} - \mathbf{v} \quad (9)$$

where $\mathbf{Q} = \int |F(\underline{\theta})|^2 \left[\mathbf{G}(\underline{\theta})\mathbf{G}(\underline{\theta})^T\right] d\underline{\theta}$. To simplify the bias expression, we approximate the sin function using a second order Taylor series, $\sin(\underline{\theta}^T \mathbf{v}) \approx \underline{\theta}^T \mathbf{v} - \frac{1}{6}(\underline{\theta}^T \mathbf{v})^3$. Using the fact that $\underline{\theta}^T \mathbf{v} = |\mathbf{v}| \underline{\theta}^T \mathbf{n}_\psi$ where $\mathbf{n}_\psi$ is the unit vector $[\cos(\psi) \sin(\psi)]^T$ we can approximate the bias function (9) as

$$\begin{aligned} \mathbf{b}(\mathbf{v}) &\approx |\mathbf{v}| \left[\mathbf{Q}^{-1} \int |F(\underline{\theta})|^2 \mathbf{G}(\underline{\theta}) \underline{\theta}^T \mathbf{n}_\psi d\underline{\theta} - \mathbf{n}_\psi\right] \\ &\quad - \frac{|\mathbf{v}|^3}{6} \int |F(\underline{\theta})|^2 \mathbf{G}(\underline{\theta}) (\underline{\theta}^T \mathbf{n}_\psi)^3 d\underline{\theta} \\ &= |\mathbf{v}| \mathbf{c}_1(\psi) - |\mathbf{v}|^3 \mathbf{c}_2(\psi) \end{aligned} \quad (10)$$

We rewrite the 2-dimensional version of (8) in polar coordinates as

$$J(\mathbf{a}_x, \mathbf{a}_y) = 2 \int_0^\pi \int_0^V \mathbf{b}(\mathbf{v})^T \mathbf{b}(\mathbf{v}) d|\mathbf{v}| d\psi \quad (11)$$

If desired, one could design an optimal filter assuming that the motion was constrained to a particular angular direction. In practice, we approximate (11) as

$$\sum_\psi \frac{1}{3} V^3 \mathbf{c}_1^T \mathbf{c}_1 + \frac{2}{5} V^5 \mathbf{c}_1^T \mathbf{c}_2 + \frac{1}{7} V^7 \mathbf{c}_2^T \mathbf{c}_2 \quad (12)$$

Thus, the costly integrals performed in computing the bias now require only one set of integrations to compute the $\mathbf{c}(\psi)$'s once for each angle ψ. This greatly improves the speed in computing the cost function $J(\mathbf{a}_x, \mathbf{a}_y)$. It is this approximation that we use to generate the 2-D gradient filters. In the section V we verify the utility of such an optimization scheme.

In practice, iterative multiscale estimation provides significant improvements in estimator accuracy [8]. The multiscale approach decomposes the pair of images into dyadic pyramids of lowpass filtered and downsampled images denoted $z_{1,2}^h(m, n)$ where the superscript h denotes the level of pyramid. This creates an image pair at the top of the pyramid to be the coarsest image of size $\frac{M}{2^h}$ by $\frac{N}{2^h}$. And, the original image sequence lies at the bottom of the pyramid. The iterative multiscale estimation begins by estimating translation between the image pair at the coarsest scale (the top of the pyramid) using the estimator of (5). After estimating the translation $\hat{\mathbf{v}}^1$ at the coarsest level, the first image at the next finer resolution level of the pyramid $z_1^{h-1}(m,n)$ is shifted according to $2\times$ the estimates $\hat{\mathbf{v}}^1$ to create a a new image pair $\check{z}_{1,2}^{h-1}(m,n)$ containing only the residual motion (bias) from the previous estimate. Then, this residual motion $\hat{\mathbf{v}}^r$ is estimated from this image pair $\check{z}_{1,2}^{h-1}(m,n)$ and an the original estimate is updated according to $\hat{\mathbf{v}}^2 = 2\hat{\mathbf{v}}^1 + \hat{\mathbf{v}}^r$. This process repeats while moving down the pyramid in a coarse to fine fashion.

The multiscale approach improves estimator performance for a variety of reasons, the most important being that the magnitude of the motion in the downsampled images will necessarily be reduced by the downsampling ratio, effectively "shrinking" $\mathbf{v}$. Noting that Figure 1 exhibits the tendency of the bias to grow unacceptably large as the magnitude of the translation increases, minimizing translation magnitude helps ensure that the initial guess is indeed close to the actual estimate improving the likelihood of convergence to an unbiased estimate.

Traditionally, the same gradient filter was applied at each level of the pyramid. The performance and rate of convergence of the multiscale method can be further improved using optimally designed bias-minimizing filters. We suggest the novel approach of using *different* gradient filters at each level of the pyramid where each filter is designed according to the bias-minimizing cost function (12). We show in the following section that using a collection of bias-minimizing filters provides superior performance in multiscale estimation.

V. EXPERIMENTS

To verify the utility the bias-minimizing cost function of (12), we compare the performance of the typical filters mentioned in Section III. with the bias-minimized filters. All of the filters were separable linear phase filters with five taps (2 coefficients). The image used in the experiments is the tree image from [1]. The image was filtered with a Gaussian low-pass filter to replicate the common practice of image pre-smoothing which has been shown to improve estimator performance [1]. While the bias-minimizing filters were designed for the range $v_x, v_y \in [-2,2]$, we restrict the experimentation to the case where $v_x = v_y \in [0,2]$ because of the bias symmetry. Figure 2 shows the magnitude of the estimator bias using these different gradient filters. As indicated by the graph, the bias of all the filters becomes severe as the magnitude of the translation increases, but the bias for the optimizing filter is minimized. In fact, for large translations the bias for the optimal filter is at least 30 percent less than the the Nestares filters and half that of the Fleet filter. The optimized filters have the coefficients $g_x = [-1.5978\ 2.6721\ 0\ -2.6721\ 1.5978]$ and $g_y = [-1.2368\ 1.9353\ 0\ -1.9353\ 1.2368]^T$. We

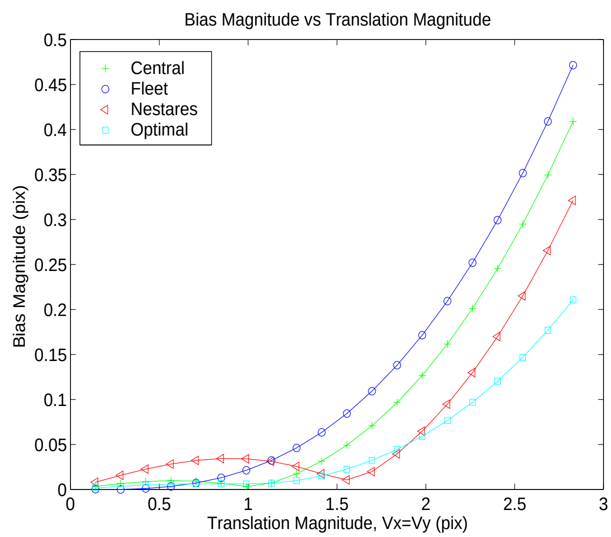

Fig. 2. Magnitude of estimator bias $|b(\mathbf{v})|$ vs translation $v_x = v_y$ for different gradient filters

found that these results reflect the general performance of the bias-minimized filters for translations other than $v_x = v_y$.

Next, to understand the effect of the bias-minimizing filters in the context of multiscale estimation, we conduct a similar experiment using multiscale gradient-based estimation. For this experiment we utilized a Gaussian pyramid with three levels designing an optimal filter for each level. All of the filters were designed for the range $v_x, v_y \in [-2,2]$. For this experiment, we restricted actual motion to the case $v_x = v_y \in [0,6]$. Figure 3 shows the magnitude of the estimator bias for the multiscale estimators displaying the capacity of the multiscale method for improving estimator performance compared with Figure 2. Overall, the bias-minimizing filters provide a dramatic improvement in estimator accuracy over the entire range of translations. In examining the results, a few observations are worth noting. While the bias functions no longer have the predictable form given by (9), the functions appear to have a certain similarity to their structure. This suggests the possibility for deriving an expression for the bias function in the multiscale estimation. Also, the biases of the Fleet and Central difference filters seem to grow rapidly at large translation magnitudes. Presumably, this effect stems from the inability to compensate for very poor coarse scale initial estimates. Overall, the bias-minimizing filters

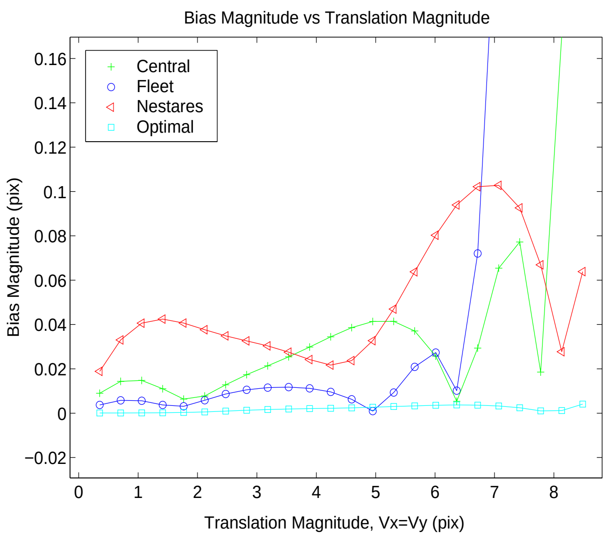

Fig. 3. Magnitude of multiscale estimator bias $|b(\mathbf{v})|$ vs translation $v_x = v_y$ for different filters

provide significant improvement over the standard filters.

VI. CONCLUSIONS AND FUTURE WORK

In our work, we have presented the fundamental relationship between gradient-based motion estimator bias and the choice of gradient filters. We have shown how this knowledge permits the design of bias-minimizing gradient filters. We have experimentally verified the utility of such filters for improving estimator performance and suggested a means of incorporating the filter design process into a multiscale estimation framework providing substantially improved estimation.

The work presented here suggests several possible directions for future work. For instance, in our experiments on multiscale methods, the filters designed at each scale might be more efficient if special attention were given to the region of operation at each level of the pyramid. Further investigation into the cost function might provide more efficient means of finding optimal filters. While the work here has focused on the high SNR regimes where bias dominates estimator error, further work might find mean square error (MSE) minimizing filters using the bounds given in [4]. Finally, we hope that this work might be extended in some fashion to improved the performance of optical flow estimation as performed in [6].

VII. REFERENCES

[1] J. Barron, D. Fleet, S. Beauchemin, and T. Burkitt, “Performance of optical flow techniques,” *CVPR*, vol. 92, pp. 236–242, 1992.

[2] J. Kearney, W. Thompson, and D. Boley, “Optical flow estimation: an error analysis of gradient-based methods with local optimization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 2, pp. 229–244, March 1987.

[3] J. W. Brandt, “Analysis of bias in gradient-based optical flow estimation,” *IEEE Asilomar Conference on Signals, Systems and Computers*, pp. 721–725, 1995.

[4] D. Robinson and P. Milanfar, “Fundamental performance limits in image registration,” *Submitted to IEEE Transactions on Image Processing*, 2003.

[5] E. Simoncelli, “Design of multi-dimensional derivative filters,” *Proceedings IEEE ICIP 1994*, 1994.

[6] M. Elad, P. Teo, and Y. Hel-Or, “On the design of optimal filters for gradient-based motion estimation,” *International Journal on Computer Vision submitted*, 2002.

[7] O. Nestares and D. Heeger, “Robust multiresolution alignment of MRI brain volumes,” *Magnetic Resonance in Medicine*, vol. 43, pp. 705–715, 2000.

[8] C. Stiller and J. Konrad, “Estimating motion in image sequences,” *IEEE Signal Processing Magazine*, vol. 16, pp. 70–91, July 1999.