Available online at www.sciencedirect.com
SCIENCE DIRECT®
Signal Processing: *Image Communication* 20 (2005) 554–568
SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

Bias minimizing filter design for gradient-based image registration

Dirk Robinson*, Peyman Milanfar

Electrical Engineering Department, University of California Santa Cruz, Santa Cruz, CA 95064, USA

Received 12 March 2005; accepted 14 March 2005

Abstract

Gradient-based image registration techniques represent a very popular class of approaches to registering pairs or sets of images. As the name suggests, these methods rely on image gradients to perform the task of registration. Very often, little attention is paid to the filters used to estimate image gradients. In this paper, we explore the relationship between such gradient filters and their effect on overall estimation performance in registering translated images. We propose a methodology for designing filters based on image content that minimize the estimator bias inherent to gradient-based image registration. We show that minimizing such bias improves the overall estimator performance in terms of mean square error (MSE) for high signal-to-noise ratio (SNR) scenarios. Finally, we propose a technique for designing such optimal gradient filters in the context of iterative multiscale image registration and verify their further improved performance.

Keywords: Motion estimation; Optical flow; Bias; Filter design

1. Introduction

The problem of image registration can be expressed as finding a spatial coordinate transformation that relates one image to another. Knowledge of this spatial transformation is required whenever multiple images of a similar scene need to be compared or combined in some sense. For example, to fuse a collection of images for the purpose of improving spatial resolution as in [6] the unknown translations between pairs of images must be estimated to a very high degree of accuracy. For this and other applications, the accuracy of any image registration algorithm must be well understood.

When the observed image data consists of a pair of noisy images, the data can be modelled as

$$z_1(m,n) = f(m,n) + \varepsilon_1(m,n),$$
$$z_2(m,n) = f(m - v_x(m,n), n - v_y(m,n)) + \varepsilon_2(m,n),$$

where $\varepsilon_i(m,n)$ is zero-mean white Gaussian noise with variance σ^2 and $\vec{v}(m,n) = [v_x(m,n), v_y(m,n)]^{\mathrm{T}}$

*Corresponding author. Tel.: +1 650 496 5703.
E-mail address: dirkr@ee.ucsc.edu (D. Robinson).

0923-5965/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2005.03.010

is the unknown vector field characterizing the transformation between the two images. In practice, such a noise model has been found to accurately capture the effects of random noise in typical imaging systems [3]. We use the notation m,n to represent the M by N sample locations $f(mT,nT)$ for an underlying continuous image $f(x,y)$. The estimation of the unknown motion vector field is also commonly referred to as motion estimation or optical flow estimation.

One very popular class of methods for estimating image motion is the so called *gradient-based* or *differential-based* methods [1]. These methods estimate image motion by relating the change in image intensity between images to spatial image gradients. As the name implies, gradient-based estimation methods require measurements of image gradients. These measurements are invariably obtained by application of simple, linear phase, shift invariant filters. Even though these filters play a vital role in the estimation scheme, and have been shown to affect estimator bias [9,2,11], relatively few researchers have studied the design of such filters [12,4,5]. While many papers acknowledge the errors incurred by such gradient approximation schemes, they treat these errors as random in nature and construct statistically robust estimators to minimize their effect. These methods have overlooked potential improvement in estimator accuracy by improving the computation of image gradients with optimally designed gradient filters.

In this paper, we use the bias formulation, presented in [11], to propose a systematic method for designing gradient filters to optimize translation estimation performance. We explore an optimization scheme whereby a gradient filter is designed based on the image under observation to minimize overall estimator bias. We detail an approach for designing such filters for multiscale iterative image registration and experimentally verify the bias-minimizing properties of such filters in both a non-iterative and multiscale iterative framework. We show that such filters improve the mean square error (MSE) performance for high signal-to-noise ratios (SNR) as well. We compare the performance of such optimized filters with previously used and proposed filters and in particular those designed using the method of [5]. We conclude by presenting future research questions relating to the problem of filter design for gradient-based motion estimation and image registration.

1.1. Gradient-based motion estimation

To motivate our filter design methodology, we first explore the gradient-based image registration technique and the review the current methods addressing filter design. For the sake of simplicity, we first present our initial analysis in 1-D. For the 1-D case, we suppose that the measured data is of the form

$$z_1(k) = f(k) + \varepsilon_1(k), \tag{1}$$

$$z_2(k) = f(k + v(k)) + \varepsilon_2(k), \tag{2}$$

where ε_i are Gaussian white noise random fields with variance σ^2 and $v(k)$ is the unknown transformation vector field. Again, we assume that the image (function) $f(x)$ is sampled above the Nyquist rate. Throughout this paper, we assume that the noise power σ^2 inherent to the imaging system has been effectively characterized prior to estimation; that is, we assume that σ^2 is known.

The family of gradient-based motion estimators begin by linearizing $f(k+v)$ about the point $v=0$ in a Taylor series as,

$$f(k + v(k)) = f(k) + v(k)f'(k) + R(k, v(k)), \tag{3}$$

where $R(k,v(k))$ denotes the approximation error of the linearization.

Most of the gradient-based methods make two simplifying assumptions about (3). First, it is assumed that the unknown vector field $v(k)$ is comprised of locally parametric vector fields. The simplest of models is the translational model of image motion where the coordinate transformation is assumed to be constant over some region in space $v(k) = v,\ \forall k \in \Omega$. Here, Ω is a local region in the image space, possibly the entire image. Second, it is assumed that the remainder term $R(k,v)$ is negligible. With such assumptions, estimation of the unknown translation v consists of solving an overdetermined set of linear equations of the form

$$z_2(k) - z_1(k) = vf'(k) + \varepsilon(k), \tag{4}$$

where each equation is defined for a particular pixel k in Ω. Here $\varepsilon(k)$ is defined as $\varepsilon(k) \equiv \varepsilon_2(k) - \varepsilon_1(k)$. [1]

In practice, a set of filters are applied to the images to generate a set of linear equations of the form (4). For instance, pre-smoothing filters are often applied to the pair of images prior to estimation to improve the accuracy of the linearized signal model. We refer the reader to [5] for a complete motivation of the use of pre-smoothing filters. The pair of images z_1 and z_2 are pre-filtered using a low pass filter to produce smoothed versions of the images; viz

$$\tilde{z}_1(k) = h(k) * z_1(k), \tag{5}$$

$$\tilde{z}_2(k) = h(k) * z_2(k). \tag{6}$$

(where $*$ represents convolution).[2] Furthermore, the gradient (derivative) $f'(k)$ must be approximated from the measured data using a gradient filter $g(k)$. The gradient filter is applied to one of the available images to approximate the image gradient as

$$\tilde{f}'(k) = g(k) * \tilde{z}_1(k). \tag{7}$$

Using this set of filters, the form of (4) is generalized as

$$\tilde{z}(k) = v\tilde{f}'(k) + \tilde{\varepsilon}(k), \tag{8}$$

where $\tilde{z}(k) \equiv \tilde{z}_2(k) - \tilde{z}_1(k)$. Here, we see that the nonlinear signal model is no longer expanded in a Taylor series, but by an alternate power series implicitly defined by the gradient filter $g(k)$. Again, the linear signal model of (8) essentially ignores the implied approximation error $\tilde{R}(k, v)$.

Given such processed image data, the least-squares (LS) estimate for translation between a pair of images in the region Ω is given by

$$\hat{v} = \frac{\sum_{k\in\Omega} \tilde{f}'(k)\tilde{z}(k)}{\sum_{k\in\Omega} (\tilde{f}'(k))^2}. \tag{9}$$

[1] In this paper, we focus on the estimation of global translational as a first step in understanding the general problem of filter design for image registration.

[2] In [5], a more general pre-smoothing approach is employed wherein *different* pre-smoothing filters $h_1(k)$ and $h_2(k)$ are applied to each of the images.

Traditionally, the choice of smoothing filters and gradient filters that were utilized have been designed heuristically. In the next section we review the earlier work relating to principled filter design for gradient-based estimation.

1.2. Filter design for gradient-based motion estimation

Very little work has been done addressing the design of filters specifically for application to motion estimation. To our knowledge, such an approach was first studied in [4] and then later in [5] which extends the generic (not necessarily application specific) gradient filter design principles of [12]. For both of these approaches, it is assumed that the gradient filter $g(k)$ is anti-symmetric (because it is a first order derivative filter) and the *pair* of pre-smoothing filters $h_1(k)$ and $h_2(k)$ are symmetric. Because the filters are symmetric and antisymmetric, only half of the filter coefficients need to be represented. We use the bold notation $\mathbf{h}$ and $\mathbf{g}$ to denote the vector of filter coefficients used to construct the filters $h(k)$ and $g(k)$, respectively. The symmetry of the pre-smoothing filters guarantees their being linear phase filters and the anti-symmetry of the gradient filter is necessary to preserve the property of being a differentiator [10].

In [12], a method for designing a single pre-smoothing filter $h(k)$ and a derivative filter $g(k)$ is proposed. The pair of filters are designed such that the analytic derivative of the smoothed signal will be similar to the approximated derivative signal using the filter $g(k)$. To do so, [12] proposes a cost function to design the filters $h(k)$ and $g(k)$. The cost function attempts to minimize the total energy in the difference signal between the analytic derivative of the pre-smoothed signal $f'(x) * h(x)$ and the filter-based derivative approximation $\tilde{f}'(x)$ weighted by the amplitude spectrum of the image function $f(x)$. The image model is assumed to have the amplitude spectrum $|F(\theta)| = \frac{1}{\sqrt{|\theta|}}$ a representative *natural* image. The cost function in [12] is expressed in the Fourier domain as

$$J_1(\mathbf{h}, \mathbf{g}) = \int_{-\pi}^{\pi} \frac{1}{|\theta|} [j\theta H(\theta) - G(\theta)]^2 \, \mathrm{d}\theta. \tag{10}$$

Here, $H(\theta)$ and $G(\theta)$ are the Fourier transforms for the desired filters given by

$$H(\theta) = \{\mathbf{h}\}_0 + 2\sum_{i=1}^{K_h}\{\mathbf{h}\}_i \cos(i\theta),$$

$$G(\theta) = 2\sum_{i=1}^{K_g}\{\mathbf{g}\}_i \sin(i\theta),$$

where $\{\mathbf{x}\}_i$ represents the ith component of a vector $\mathbf{x}$. The formulation of (10) reflects the fact that the Fourier transform of a derivative signal $f'(x)$ is given by $j\theta F(\theta)$ [10]. Such a design philosophy attempts to minimize approximation error due to the application of FIR filters to estimate image gradients. In [12], the solution was found by formulating the optimization problem as an eigenvalue problem. Several filter pairs of differing lengths are presented in the paper. While [12] does not directly address the application of such filters to estimate motion, the filters have been noted to improve estimator performance [2].

In [5], these filter design principles were extended to address the specific problem of gradient-based motion estimation. By designing a *set* of pre-smoothing filters and gradient filters minimizing the modelling error for a particular image, the estimator performance could be further improved. This alternate design procedure can be expressed as that of finding the filter coefficients for the set of filters which minimize the remainder term $\tilde{R}(k,v)$ of the linearized model. The goal of such an approach is that of improving the quality of the motion estimates by minimizing the approximation error. The authors derive a cost function taking into account a specific image as well as a range of possible translations $v \in [-V, V]$. Such a cost function has the form

$$\begin{aligned} J_2(\mathbf{h}_1,\mathbf{h}_2,\mathbf{g}) &= \int_{-V}^{V}\int_{-\pi}^{\pi} |F(\theta)|^2 |e^{j\theta v}H_2(\theta) \\ &\quad - H_1(\theta) - vG(\theta)|^2 \,d\theta\,dv \\ &= \int_{-V}^{V}\int_{-\pi}^{\pi} |F(\theta)|^2 \big|\Upsilon(\theta)\big|^2 \,d\theta\,dv. \end{aligned} \tag{11}$$

Intuitively, the filter designed should minimize the energy in the modelling error $\tilde{R}(k,v)$ weighted by the image spectrum over a given range of unknown translations. The authors note that minimizing the error alone will not provide good filters since the optimization tends to create "non-informative" filters which contain most of their spectral energy at frequencies where the image spectral energy $|F(\theta)|^2$ is lowest. They correct this by adding an additional penalty term balancing the desire to tune the filter to the given image with that of an image with a flat spectrum. This modified cost function looks like

$$\begin{aligned} &J_2(\mathbf{h}_1,\mathbf{h}_2,\mathbf{g}) \\ &\quad = \int_{v=-V}^{V}\int_{-\pi}^{\pi} [\alpha + (1-\alpha)|F(\theta)|^2]|\Upsilon(\theta)|^2 \,d\theta\,dv, \end{aligned} \tag{12}$$

where α is a tuning parameter to be applied during the filter design process. The authors also find a solution to this problem by again solving an eigenvalue problem. To date, the proposed method represents the only work addressing filter design specifically for the problem of motion estimation.

While these previous works have made fundamental contributions to gradient-based motion estimation, they ignore the particular structure of the gradient-based motion estimator which ultimately characterizes the statistical performance of such estimators. In this paper, we use the statistical performance of the estimator to guide the design process. Specifically, we present a scheme for designing filters which minimize the bias of the registration algorithm.

2. Designing bias-minimizing filters

It has been noted on numerous occasions in the past [9,2,11] that gradient-based estimators produce biased estimates. However, only recently in [11], we derived the explicit form of this bias. For completeness, we now summarize the derivation of this bias originally presented in [11].

The estimator bias as defined in [8], is given by,

$$\begin{aligned} b(v) &\equiv E[\hat{v}] - v \\ &= E\left[\frac{\sum_{k\in\Omega}\tilde{f}'(k)\tilde{z}(k)}{\sum_{k\in\Omega}(\tilde{f}'(k))^2}\right] - v \\ &= E\left[\frac{\sum_{k\in\Omega}\tilde{f}'(k)[v\tilde{f}'(k) + \tilde{R}(v,k) + \tilde{\varepsilon}(k)]}{\sum_{k\in\Omega}(\tilde{f}'(k))^2}\right] - v \\ &= E\left[\frac{\sum_{k\in\Omega}\tilde{f}'(k)[\tilde{R}(v,k) + \tilde{\varepsilon}(k)]}{\sum_{k\in\Omega}(\tilde{f}'(k))^2}\right], \end{aligned} \tag{13}$$

where the expectation is taken with respect to the noise vector ε. Here, we see that the bias has a component dependent on the random additive noise ε as well as the deterministic approximation error $\tilde{R}(v,k)$. In [11], it was shown that for high SNR situations, the bias is dominated by deterministic modelling error. For many computer vision and image registration applications, the effective SNR falls into this high SNR regime. This deterministic bias results from the approximation error in the data model $\tilde{R}(v,k)$ due to the linear signal approximation. In [11] we showed that this estimator bias can be expressed in the Fourier domain as

$$b(v) \approx \frac{\int_{-\pi}^{\pi} |\tilde{F}(\theta)|^2 [G(\theta)\sin(v\theta)]\,\mathrm{d}\theta}{\int_{-\pi}^{\pi} |\tilde{F}(\theta)G(\theta)|^2\,\mathrm{d}\theta} - v. \tag{14}$$

By way of the substitution $\tilde{F}(\theta) = F(\theta)H(\theta)$ we can rewrite (14) as

$$b(v) \approx \frac{\int_{-\pi}^{\pi} |F(\theta)|^2 |H(\theta)|^2 [G(\theta)\sin(v\theta)]\,\mathrm{d}\theta}{\int_{-\pi}^{\pi} |F(\theta)H(\theta)G(\theta)|^2\,\mathrm{d}\theta} - v. \tag{15}$$

It is this bias function that we use to develop a filter design methodology. We refer the interested reader to [11] for the complete derivation of this estimator bias function.

In (15) we see that the bias depends on three factors: the image content f, the choice of filters $g(k)$ and $h(k)$, and the unknown translation v. Using this bias function we construct the following cost function for finding the filter coefficients:

$$J(\mathbf{g},\mathbf{h}) = \int w(v) b^2(\mathbf{g},\mathbf{h})\,\mathrm{d}v, \tag{16}$$

where $w(v)$ is a weighting function over the space of translations v. For instance, $w(v)$ may reflect a prior distribution on the unknown translation. Such a cost function captures the desired goal of minimally biased estimates of image translation. For our purposes, we assume that the translations are equally likely to be any value in a particular range $v \in [-V, V]$. In other words, we focus on the case where the weighting function is given by

$$w(v) = \begin{cases} \dfrac{1}{2V}, & v \in [-V, V], \\ 0 & \text{else.} \end{cases} \tag{17}$$

Such a weighting function penalizes estimator bias equally for all translations within a particular range.

We now explore a simple method for minimizing such a cost function. Because of the nonlinear dependence on $G(\theta)$ and $H(\theta)$ (and hence $\mathbf{g}$ and $\mathbf{h}$) in the bias function (15), we propose focussing on the design of the gradient filter coefficients $\mathbf{g}$. While it would be possible to efficiently minimize (16) in a cyclic coordinated descent type algorithm which alternates between optimizing over $\mathbf{g}$ and $\mathbf{h}$, we have found in practice that optimizing both filters does not offer significantly improved performance over optimizing the gradient filter alone. Our experiments suggest that after optimizing the gradient filter, the cost function effectively finds a relatively flat region of local minima where small perturbations of the pre-smoothing filter fail to improve overall performance. As such, we first choose a pre-smoothing filter, and then optimize over only the gradient filter. Here we present the algebraic simplifications useful for highly efficient filter optimization. We note that similar simplifying operations are applicable for the 2-D case as well.

First, we rewrite the bias function (15) in vector form as

$$b(v) = \frac{\mathbf{s}(v)^{\mathrm{T}}\mathbf{W}\mathbf{D}\mathbf{g}}{\mathbf{g}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{W}\mathbf{D}\mathbf{g}} - v, \tag{18}$$

where

$$\{\mathbf{s}(v)\}_a = \sin(v\theta_a),$$

$$\{\mathbf{W}\}_{a,b} = \begin{cases} |F(\theta_a)|^2 |H(\theta_a)|^2, & a = b \\ 0, & a \neq b, \end{cases}$$

$$\{\mathbf{D}\}_{a,k} = \sin(k\theta_a).$$

In these equations, the a indicates the spatial frequency used in the DFT such that $\theta_a = \pi - \frac{a2\pi}{K}$.

For a fixed pre-smoothing filter, the cost function $J_g(\mathbf{g})$ can be written in vector form as

$$\begin{aligned} J_g(\mathbf{g}) &= \int_{-V}^{V} b^2(\mathbf{g})\,\mathrm{d}v \\ &= \int_{-V}^{V}\left[v^2 + \frac{\mathbf{g}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{s}(v)\mathbf{s}(v)^{\mathrm{T}}\mathbf{W}\mathbf{D}\mathbf{g}}{(\mathbf{g}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{W}\mathbf{D}\mathbf{g})^2} \right. \\ &\quad \left. -2v\,\frac{\mathbf{s}(v)^{\mathrm{T}}\mathbf{W}\mathbf{D}\mathbf{g}}{\mathbf{g}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{W}\mathbf{D}\mathbf{g}}\right]\mathrm{d}v \\ &= \frac{2V^3}{3} + \frac{\mathbf{g}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\tilde{\mathbf{S}}\mathbf{W}\mathbf{D}\mathbf{g}}{(\mathbf{g}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{F}\mathbf{D}\mathbf{g})^2} \\ &\quad - 2\,\frac{\mathbf{p}^{\mathrm{T}}\mathbf{F}\mathbf{D}\mathbf{g}}{\mathbf{g}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{F}\mathbf{D}\mathbf{g}}, \end{aligned} \tag{19}$$

where

$$\begin{aligned} \{\tilde{\mathbf{S}}\}_{a,b} &= \int_{-V}^{V} \sin(v\theta_a)\sin(v\theta_b)\,\mathrm{d}v \\ &= \frac{2\sin(V(\theta_a-\theta_b))}{\theta_a-\theta_b} - \frac{2\sin(V(\theta_a+\theta_b))}{\theta_a+\theta_b} \end{aligned} \tag{20}$$

and

$$\begin{aligned} \{\mathbf{p}\}_a &= \int_{-V}^{V} v\,\sin(v\theta_a)\,\mathrm{d}v \\ &= \frac{2\sin(V\theta_a) - 2V\theta_a\cos(V\theta_a)}{\theta_a^2}. \end{aligned}$$

It is the simple closed form for such integrals which makes such an optimization simple to implement. While not obvious, it is important to note that matrix $\tilde{\mathbf{S}}$ in (20) represents a convolution operation because of the spectral symmetry of $|F(\theta)H(\theta)|^2G(\theta)$ about $\theta = 0$. Thus, the left multiply by the matrix $\tilde{\mathbf{S}}$ can be implemented using FFT operations thereby removing the necessity of constructing the large matrix $\tilde{\mathbf{S}}$. Such implementation becomes critical for the 2-D scenario where the dimensions of the matrices are much larger.

By defining

$$\begin{aligned} \mathbf{Q}_1 &= \mathbf{D}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\tilde{\mathbf{S}}\mathbf{W}\mathbf{D}, \\ \mathbf{Q}_2 &= \mathbf{D}^{\mathrm{T}}\mathbf{F}\mathbf{D}, \\ \mathbf{q} &= \mathbf{p}^{\mathrm{T}}\mathbf{F}\mathbf{D}, \end{aligned}$$

we rewrite (19) as

$$J_g(\mathbf{g}) = \frac{2V^3}{3} + \frac{\mathbf{g}^{\mathrm{T}}\mathbf{Q}_1\mathbf{g}}{(\mathbf{g}^{\mathrm{T}}\mathbf{Q}_2\mathbf{g})^2} - \frac{\mathbf{q}^{\mathrm{T}}\mathbf{g}}{\mathbf{g}^{\mathrm{T}}\mathbf{Q}_2\mathbf{g}}. \tag{21}$$

The matrices $\mathbf{Q}_{1,2}$ and the vector $\mathbf{p}$ need to be computed only once during the optimization, greatly simplifying the overall computational complexity. The nonlinearity of the cost function becomes immediately apparent in the form of the cost function. Because the dimensions of the filters of interest are relatively small (2–4 unique coefficients), we utilize a black box Matlab optimization routine `fminunc` to perform the optimization. In our experiments, we use a standard filter such as the Fleet filters [1] as an initial guess for the optimization routine.

3. Filter design for 2-D multiscale iterative registration

One important generalization of the filter design methods of [12] and [5] and that proposed in the previous section, is the extension to the design of 2-D filters. Both of these previous methods have addressed only the filter design problem for the 1-D case. The extension to the 2-D case for [5] involves designing *generic* (non image-specific) 1-D filters. In [5], *generic* 1-D filters were designed by setting $\alpha = 1$, which essentially designs filters assuming that the image has a flat or constant spectrum. Then, the authors of both [12] and [5] describe methods for applying these optimized 1-D filters to a 2-D image which is necessary for subsequent motion estimation.

In our case, we also assume that the 2-D filters are simple 1-D filters applied in a 2-D fashion. For example, the 2-D gradient filters $g_1(m,n)$ and $g_2(m,n)$ (for the x and y dimensions) are defined as

$$g_1(m,n) = g_x(m)\delta(n), \tag{22}$$

$$g_2(m,n) = \delta(m)g_y(n), \tag{23}$$

where $g_x(k)$ and $g_y(k)$ are a pair of optimized 1-D gradient filters and

$$\delta(k) = \begin{cases} 1, & k = 0, \\ 0 & \text{else.} \end{cases} \tag{24}$$

We show how the design of such 2-D filters is a natural extension of the 1-D case presented in the previous section. Unlike [5], however, our filters are designed taking into account the 2-D spectrum of a given image instead of a generic image spectrum. In addition, we propose a methodology for designing filters for multiscale iterative image registration. We note that using such simple 1-D filters is very common in gradient-based motion estimation due to their efficiency of applying separable filters.

3.1. Filter design for 2-D registration

Using vector notation, we can represent the 2-D version of the gradient-based estimator (9) as

$$\hat{\mathbf{v}} = \mathbf{A}^{-1}\mathbf{y}, \tag{25}$$

where

$$\mathbf{A} = \sum_{m,n\in\Omega} \tilde{\nabla} f(m,n)\tilde{\nabla} f(m,n)^{\mathrm{T}},$$

$$\mathbf{y} = \sum_{m,n\in\Omega} \tilde{\nabla} f(m,n)\tilde{z}(m,n),$$

where $\tilde{\nabla}$ represents the filter approximation of the gradient operator and $\mathbf{v} = [v_x, v_y]^{\mathrm{T}}$. In this case the filters used to approximate the image gradient operator $\tilde{\nabla}$ are represented as $g_1(m,n)$ and $g_2(m,n)$. Here, we make the assumption that the image has enough spectral content so that the matrix $\mathbf{A}$ is invertible. When $\mathbf{A}$ is not invertible (well-known *aperture effect* [1]), the estimation problem is ill-conditioned for *any* gradient filter.

Again using the vector notation $\underline{\theta} = [\theta_1, \theta_2]^{\mathrm{T}}$ (here $\theta_{1,2}$ represents the spatial frequencies in the 2 dimensions) and $\mathbf{G}(\underline{\theta}) = [G_1(\underline{\theta}), G_2(\underline{\theta})]^{\mathrm{T}}$ we can express the 2-D gradient-based estimator bias as

$$\mathbf{b}(\mathbf{v}) = \mathbf{A}^{-1}\int |H(\underline{\theta})F(\underline{\theta})|^2\mathbf{G}(\underline{\theta})\sin(\underline{\theta}^{\mathrm{T}}\mathbf{v})\,\mathrm{d}\,\underline{\theta} - \mathbf{v}, \tag{26}$$

where $\mathbf{A} = \int |H(\underline{\theta})F(\underline{\theta})|^2[\mathbf{G}(\underline{\theta})\mathbf{G}(\underline{\theta})^{\mathrm{T}}]\,\mathrm{d}\,\underline{\theta}$.

As in the 1-D case, we use a cost function with a uniform weighting function $w(\mathbf{v})$ to generate

$$J_g(\mathbf{g}_x, \mathbf{g}_y) = \int \mathbf{b}(\mathbf{v})^{\mathrm{T}}\mathbf{b}(\mathbf{v})\,\mathrm{d}\mathbf{v}, \tag{27}$$

which we use design our 2-D filters according to (22) and (23). While somewhat tedious, the same algebraic simplifications apply to $J_g(\mathbf{g}_x, \mathbf{g}_y)$ as those shown in Section 2. For all of our experiments, we assume that the pre-smoothing filter is a truncated 7-tap sampled Gaussian filter of the form

$$h(x,y) = \frac{1}{\rho^2 2\pi}\,\mathrm{e}^{-\frac{x^2+y^2}{2\rho^2}}. \tag{28}$$

Here, the parameter ρ defines the width or cutoff frequency of the radially symmetric low-pass filter.

Before we begin analyzing the performance of such specially tuned filters, we examine the frequency response of the filters produced by our proposed design methodology. First, we examine the effect of the translation range $[-V, V]$ on the designed filters. Fig. 2 shows the different filters produced by our optimization scheme for different translation ranges $[-V, V]$ assuming the Tree image of Fig. 1. The pre-smoothing filter was set with a width parameter of $\rho = \sqrt{3}$ (Figs. 1 and 2). The actual filter coefficients of the optimized filters are shown in Fig. 3. We note that for several ranges of translation, these filters no longer resemble gradient filters. Instead, we see that the optimized filters exhibit strong bandpass characteristics. In fact, we observe that as the range of translations grows, the filters tend to exhibit a shift in the bandpass region towards the higher spatial frequencies. Such behavior satisfies the intuition that as the translation parameter $\mathbf{v}$ increases in magnitude, the Taylor expansion used to linearize the nonlinear signal model becomes less accurate. The optimal filters find an alternate linearization which improves the accuracy of estimation.

Next, we examine the role the underlying image function $f(x,y)$ plays in the filter design process. As an example, Fig. 4 shows the spectral responses for filters optimized for the four images in Fig. 1 assuming a translation range $[-2, 2]$. We note that the filters' spectral responses do not appear to vary widely for the different images. This suggests that generic filters might be designed which still offer improved overall performance. For instance, we may choose the representative *natural* power

spectrum for the image [7]

$$|F(\underline{\theta})|^2 = \frac{1}{\theta_1^2 + \theta_2^2}. \tag{29}$$

Another, less generic, approach involves designing filters for a *set* of images. In this case, the filter is designed for a single image which is the average over a set of images. In our experiments, we construct an *average* image by averaging three of the four images shown in Fig. 1 with their respective mean values removed. The three images averaged were the Tree, Face, and MRI images. The DC image was intentionally left out of the averaging process to evaluate the performance of the average filter in a leave-one-out experiment. The filter coefficients designed for the average image are shown in Fig. 6 for several translation ranges. When designing a filter using the natural and average power spectra, we see that the designed filters are not very different from the image-specific optimal filters. The filters designed for the natural and average image are shown in Fig. 4 as the curve with plotted points. We see that both *natural* and *average* filters are similar to the optimized filters.

Fig. 1. Tree, DC, MRI, and Einstein images.

We note, however, that the relative independence of the optimized filters on the image spectrum depends on the range of translations $[-V, V]$. For example, Fig. 5 shows similar plots as Fig. 4 for the same images using the ranges defined by $V = 1$ and 0.25. Here, we see that for $V = 1$, the filters exhibit increases variability across the different images. Whereas, for the smaller range $V = 0.25$ all of the optimized filters are practically identical. We have observed the general behavior that for very large and very small translations ranges the optimized filters vary only slightly across different images. Around the range of $V = .5 \ldots 1.5$, however, the filters tend to

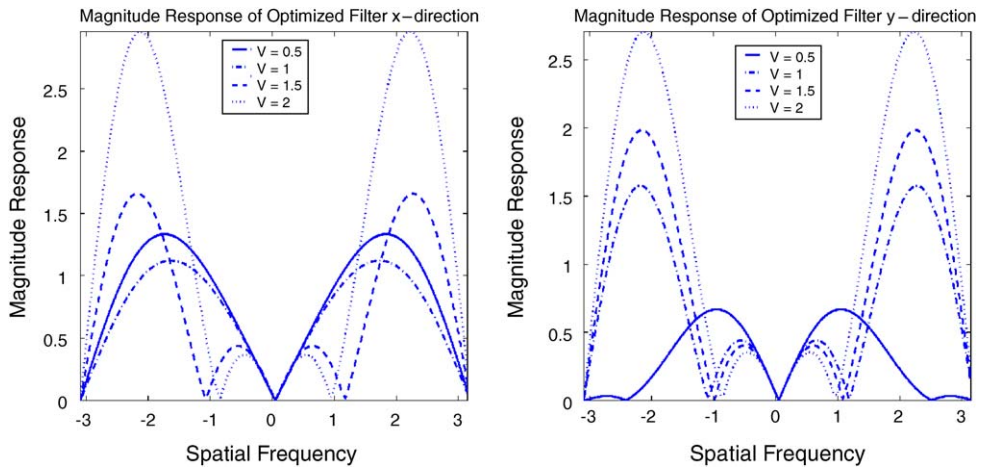

Fig. 2. Plot of the proposed filter magnitude spectral response $|G_x(\theta)|$ (left) and $|G_y(\theta)|$ (right) for different translation ranges $[-V, V]$ optimized for the Tree image.

	$g_x(k)$					$g_y(k)$				
$V = 0.5$	-0.076	0.650	0	-0.650	0.076	0.151	0.235	0	-0.235	-0.151
$V = 1.0$	-0.031	0.558	0	-0.558	0.031	0.483	-0.405	0	0.405	-0.483
$V = 1.5$	0.500	-0.437	0	0.427	-0.500	0.558	-0.569	0	0.569	-0.558
$V = 2.0$	0.754	-0.941	0	0.941	-0.754	0.697	-0.853	0	0.853	-0.697

Fig. 3. Optimized filter coefficients for the Tree image.

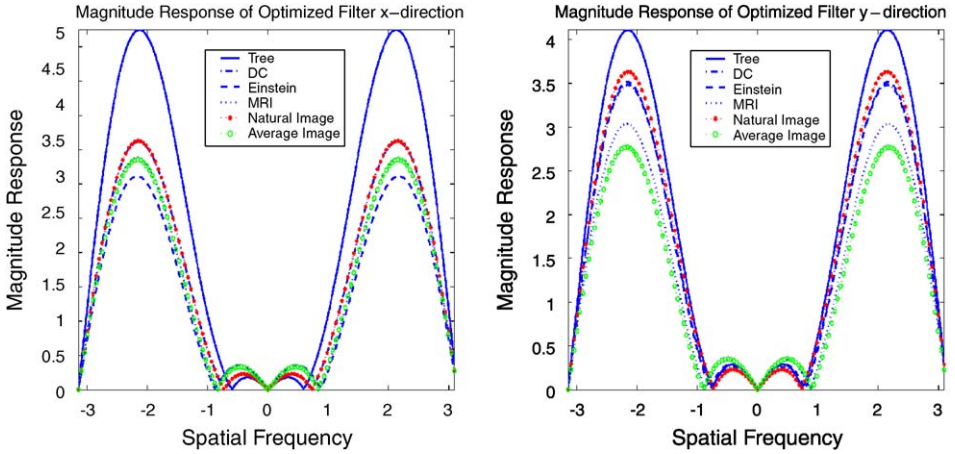

Fig. 4. Plot of the proposed filter magnitude spectral response $|G_1(\theta_1)|$ and $|G_2(\theta_2)|$ for different images optimized assuming $V = 2$.

exhibit a wider degree of variability across different images. What remains to be quantified is the reduction in performance by using these generic filters as opposed to the image-specific filters (Fig. 6).

To give an example of the overall performance improvement offered by our proposed filter design methodology, we compare the registration error over the range of translations $v_x, v_y \in [-2, 2]$. We measure the overall performance by averaging the magnitude of the registration error using for the filter sets over a set of translations in this test range. More specifically, the performance measure is given by

$$\overline{Err} = \frac{1}{N_S} \sum_{\mathbf{v} \in S_{\mathbf{v}}} \sqrt{\mathrm{MSE}(\mathbf{v})}, \tag{30}$$

where $S_{\mathbf{v}}$ is the set of test translation points of size N_S. We choose this performance measure as it shows overall performance error in units of pixels.[3] In this equation, the MSE term is defined as

$$\mathrm{MSE}(\mathbf{v}) = E[(\hat{\mathbf{v}} - \mathbf{v})^{\mathrm{T}}(\hat{\mathbf{v}} - \mathbf{v})], \tag{31}$$

where, in practice, the expectation is simulated as the sample average taken over Monte Carlo (MC) simulations for a particular value of $\mathbf{v}$. If $\mathrm{SNR} = \infty$ (no noise added to the pair of images), then the MSE depends only on the bias as $\mathrm{MSE}(\mathbf{v}) = \|\mathbf{b}(\mathbf{v})\|^2$. In a sense, the overall error is a measure of the average magnitude error over a range of translations. For our experiments, we assume that the range of translations is uniformly sampled.

We first examine the zero-noise case where $\mathrm{SNR} = \infty$. Such a scenario corresponds to the typical experimental setup examined in gradient-based estimation literature where rarely is noise

[3]We have verified that the optimized filters also show improved performance using the mean angular error performance measure of [1].

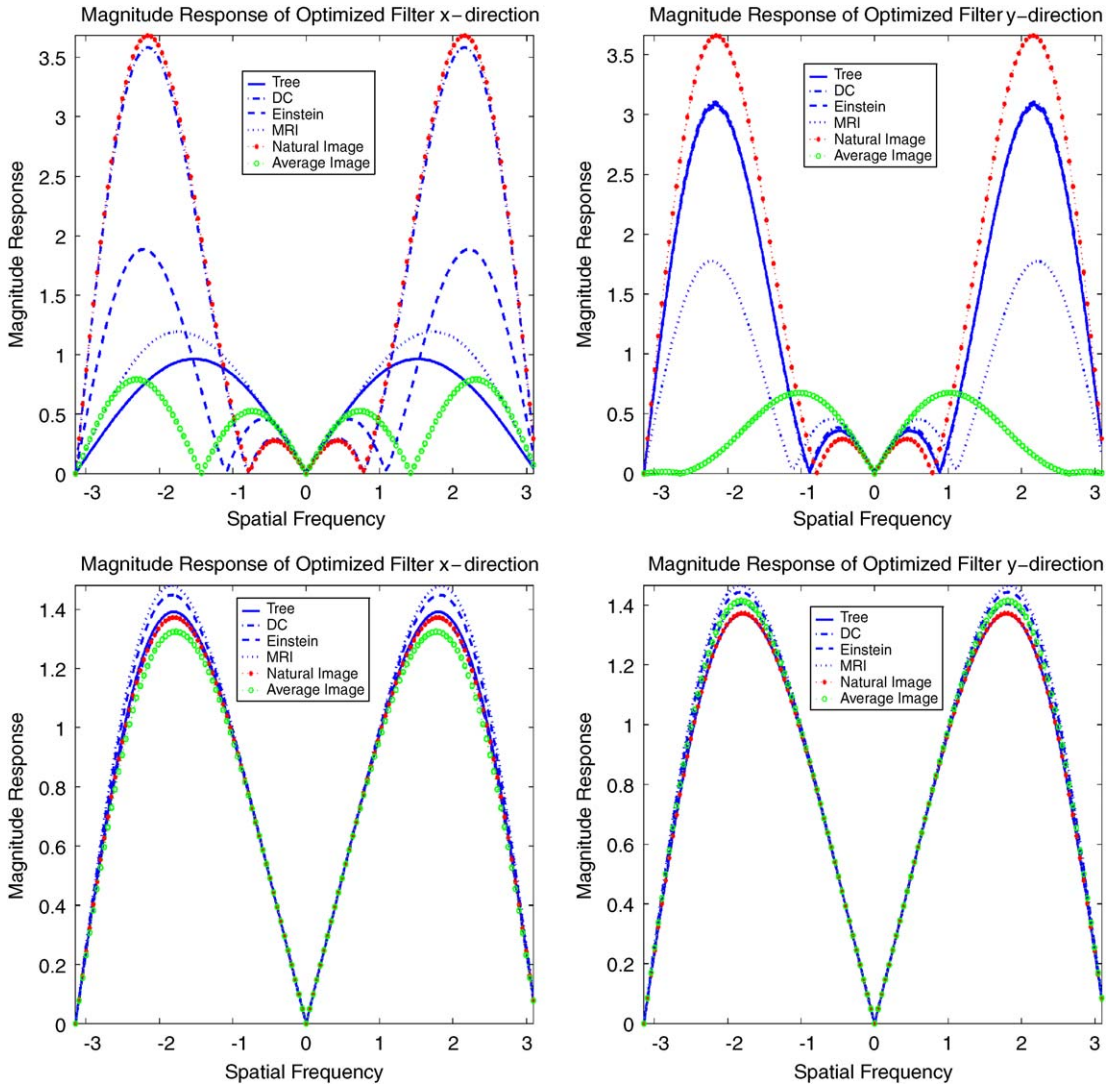

Fig. 5. Plot of proposed filter magnitude spectral response $|G_1(\theta_1)|$ and $|G_2(\theta_2)|$ for different images optimized assuming $V = 1$ (top) and $V = 0.25$ (bottom).

	$g_x(k)$					$g_y(k)$				
V =0.5	0.009	0.498	0	-0.498	-0.009	-0.049	0.603	0	-0.603	0.049
V =1.0	0.271	0.005	0	-0.005	-0.271	0.138	-0.249	0	-0.249	-0.138
V =1.5	0.587	-0.608	0	0.608	-0.587	0.455	-0.362	0	0.362	-0.455
V =2.0	0.939	-1.322	0	1.322	-0.939	0.802	-1.059	0	1.059	-0.802

Fig. 6. Optimized filter coefficients for the *average* image.

added to the images prior to estimation [1,5]. Under such conditions, only the deterministic estimator bias affects the overall estimator performance. For our experiment, we uniformly sampled the region $[-2,2]\times[-2,2]$ in increments of $[\frac{1}{10},\frac{1}{10}]$ pixels to generate our test set ($S_{\mathbf{v}}$) of translations. The filters compared were the simple central difference filter (Central), the 2nd order derivative filter mentioned in [1] (Fleet), the pair of optimized 5-tap filters from [12] (Simoncelli), the set of filters designed using the method of [5] (Elad) and the generic filter designed using our proposed method with the natural spectrum of (29). All of the filters have 5 taps (2 coefficients) except the filters of [5] which were 9 tap filters.[4] Prior to estimation, the images were pre-filtered either with the (7-tap) pre-smoothing filter of (28) with $\rho=\sqrt{3}$ or the specially tuned filters of [12,5]. We note that the spectral cutoff of such a filter with $\rho=\sqrt{3}$ is very similar to that of the binomial filter suggested in [1]. The performances for each filter set are shown in the table in Fig. 7 for the images in Fig. 1. The optimized filter shows improved overall performance for all images except for the MRI image, where the optimized filter performance was only slightly worse than that of Elad [5]. Recalling that the Elad filters were 9-tap filters as opposed to the 5-tap optimized filters, we see that, in general, the proposed filters improve average estimator performance while realizing computational savings. Furthermore, we found that when using larger optimized filters, we can achieve even greater improvement over the other filters. This improved performance results from the increased degrees of freedom of the optimization routine, where larger filters allow for more precision in tuning the frequency response of the filters. We shall show this momentarily. Finally, we note that the generic filter designed using the *natural* spectrum offer inferior performance. The filters optimized for the *average* image, however, offer moderate improvement over the other standard filters. The improved performance even extends to the DC image which was left out of the average image. This suggests that the average image has a much better representative spectrum than the model of the natural image.

[4]Filters designed using the method of [5] with less than 9 taps exhibited very poor performance.

	Tree	DC Sat.	MRI	Einstein
Central	0.111	0.162	0.145	0.121
Fleet	0.146	0.213	0.190	0.165
Simoncelli	0.069	0.103	0.094	0.072
Elad	0.112	0.074	0.056	0.063
Natural	0.3912	0.139	0.241	0.1265
Average	0.074	0.082	0.084	0.049
Optimized	0.050	0.046	0.059	0.043

Fig. 7. Overall registration error $\overline{Err}$ for the range $(v_x,v_y)\in[-2,2]\times[-2,2]$.

To visualize the effect of the optimized filters, Fig. 8 shows the bias magnitude $\|\mathbf{b}(\mathbf{v})\|$ for the Tree image. The top graph shows the bias magnitude when the [12] (Simoncelli) filters were used (the second best performance). The bottom graph shows the bias magnitude when using the filters designed by minimizing (27).

From bias exhibited in Fig. 8, we see that the bias magnitude primarily depends on the magnitude of the translation $\|\mathbf{v}\|$. In other words, the registration error is most severe along the line $v_x=v_y$. Because of this, when visually comparing the performance of different filters, we show the registration error over a range of translations along the line $v_x=v_y$. This representative slice, reveals the important performance characteristics of each filter set. For example, Fig. 9 compares the bias magnitude $\|\mathbf{v}\|$ for all of the filters when registering the Tree image. Here, we see that while the bias magnitude of all the filters becomes severe as the magnitude of the translation increases, the bias for the optimizing filter is minimized. We remind the reader that for certain ranges of velocities, filters other than the optimized filters may produce better estimates, but averaging over the entire range of translations, the optimal filter is superior. Fig. 9 also shows the performance when a 9-tap optimized filter is used. Here, we see that the larger filter shows slightly improved performance over the 5-tap optimized filter. Again, both

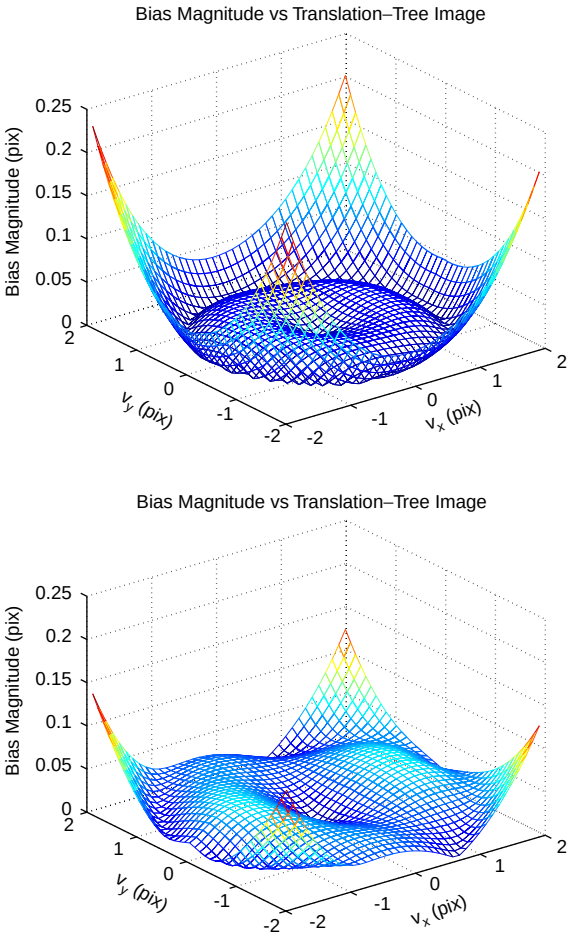

Fig. 8. Magnitude of estimator bias $\|\mathbf{b}(\mathbf{v})\|$ vs translation using the Simoncelli filters (top) and the bias minimizing gradient filters (bottom).

filters were optimized for the translation range $V = 2$.

To evaluate the performance of the optimized filter in a more realistic scenario, we must compare estimator performance in the presence of noise. To this end, we conduct Monte Carlo (MC) simulations at SNR ranging from about 10 dB through 60 dB.[5] At each SNR, we measure the MSE in estimating $\mathbf{v}$ along the line $v_x = v_y \in [0, 2]$ in increments of $\frac{1}{10}$ pixels by averaging the square estimator error over 1000 MC runs. As before, we

[5]The SNR is measured as $\mathrm{SNR} = 10\log_{10}\frac{\sigma_f^2}{\sigma^2}$ where σ_f^2 is the variance of the clean frame and σ^2 the variance of the noise.

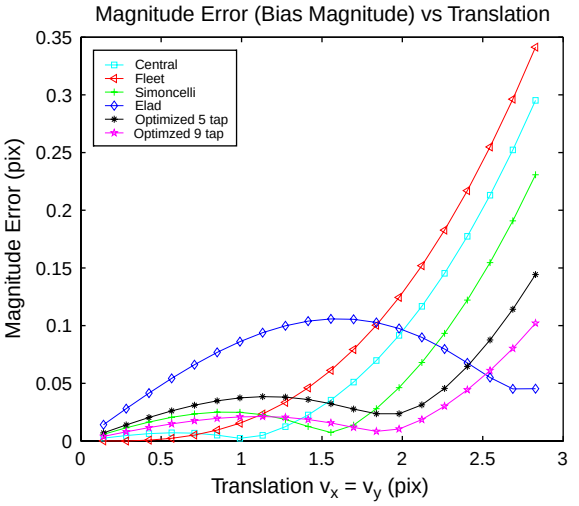

Fig. 9. Magnitude of estimator bias $\|\mathbf{b}(\mathbf{v})\|$ vs translation magnitude $\|\mathbf{v}\|$ where $v_x = v_y$.

use the same experimental setup used to produce Fig. 9 in terms of filter sets. Here, we see that the optimized filters continue to outperform the other filters over the wide range of SNR. We note that the performance does not vary widely until relatively low SNR (12 dB), as the bias dominates the MSE, for higher SNR as shown in [11]. Essentially, Fig. 10 shows that the optimized filters retain their competitive performance over a wide range of imaging SNR. We next examine similar questions in the multiscale framework.

3.2. Filter design for multiscale iterative registration

In practice, it has been noted that iterative multiscale estimation provides significant improvements in estimator accuracy [13]. The multiscale approach decomposes the pair of images into dyadic pyramids of lowpass filtered and downsampled images denoted $z_{1,2}^l(m,n)$ where the superscript l denotes the level of pyramid. This creates an image pair at the top of the pyramid to be the coarsest image of size $\frac{M}{2^l}$ by $\frac{N}{2^l}$. The original M by N images lie at the bottom of the pyramid.

Iterative multiscale registration begins by estimating translation between the image pair at the coarsest scale (the top of the pyramid) using (25).

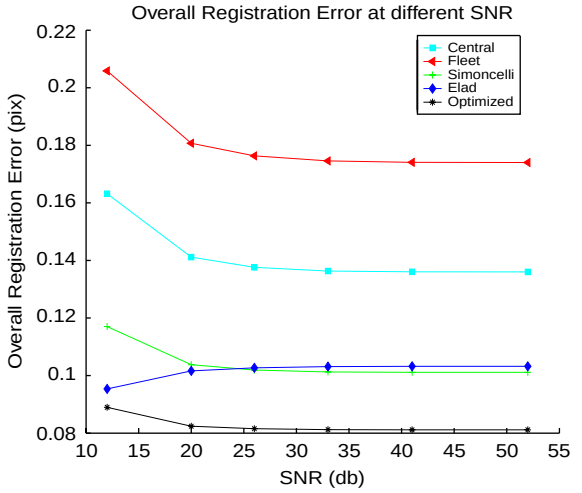

Fig. 10. Overall estimation error $\overline{Err}$ at different SNR over $v_x = v_y \in [0, 2]$ for the Tree image.

After estimating the translation $\hat{\mathbf{v}}^1$ at the coarsest level, the first image at the next finer resolution level of the pyramid $z_1^{l-1}(m,n)$ is shifted according to $2\times$ the estimates $\hat{\mathbf{v}}^1$ to create a new image pair $z_{1,2}^{l-1}(m,n)$ containing only the residual motion (dominated by bias for high SNR) from the previous estimate. Then, this residual motion $\hat{\mathbf{v}}^r$ is estimated from this image pair $z_{1,2}^{l-1}(m,n)$ and an original estimate is updated according to $\hat{\mathbf{v}}^2 = 2\hat{\mathbf{v}}^1 + \hat{\mathbf{v}}^r$. Here, the factor of 2 accounts for the effect of downsampling. 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. Most importantly, the magnitude of the motion in the downsampled images will necessarily be reduced by the downsampling ratio, effectively "shrinking" $\mathbf{v}$. Fig. 9 exhibits the tendency of the bias to grow worse as the magnitude of the translation increases. Thus, keeping the translation magnitude "artificially" small helps ensure that the estimate has a better chance of having small bias. Finally, if the bias of the initial (coarse) estimate is small, then the bias inherent to each subsequent iterations will shrink, converging to an estimate with less error. Our experiments verify this behavior.

Traditionally, the same gradient filter has been applied at each level of the pyramid. We demonstrate that 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 designing *different* gradient filters for each level of the pyramid where each filter is designed according to the cost function (27). Optimizing gradient filters in such a manner improves the convergence rates of the iterative estimation by reducing the residual motion left over from biased estimates produced from earlier iterations. More importantly, minimizing estimator bias reduces the possibility of the iterative estimation process from diverging, thereby offering a more stable method of estimation. Furthermore, since at every iteration the residual motion to be estimated is made smaller, we propose designing filters which assume that the ranges of translation shrink as the iterations proceed down the pyramid.

To show an example of such optimized filters for the multiscale registration scenario, we design gradient filters for a three level multiscale pyramid. As in Section 3.1, we first examine the zero-noise scenario ($\mathrm{SNR} = \infty$) where only the bias contributes to estimator MSE. The optimized gradient filters were designed for the translation ranges $v_x, v_y \in [-2, 2], [-.5, .5], [-.2, .2]$ for each of the three pyramid levels. Fig. 11 shows the overall multiscale registration error over the translation test set $v_x, v_y \in [-6, 6]$ uniformly sampled with a spacing of $[\frac{1}{5}, \frac{1}{5}]$ pixels. Again, we see that the optimized filters offer superior performance for multiscale estimation in terms of the registration error over a wide range of translations.

As before, to visualize the estimator performance in the multiscale setting, the registration error for the Tree image is plotted in Fig. 12 along the line $v_x = v_y \in [0, 6]$ for the zero-noise scenario. While all of the estimators show significant improvement over the non-multiscale iterative approach, the bias-minimizing 5-tap filters offer consistent improvement in estimator accuracy over the entire range of translations. For practical applications, the registration error is so small as to be considered almost unbiased. Overall, we see that principled filter design offers improvement for multiscale image registration as well.

	Tree	DC Sat.	MRI	Einstein
Central	0.006	0.010	0.004	0.012
Fleet	8.14e-4	0.002	0.001	0.008
Simoncelli	0.012	0.018	0.011	0.020
Elad	0.010	0.006	0.001	0.015
Optimized	2.07e-4	5.57e-4	2.57e-4	0.006

Fig. 11. Overall registration error $\overline{Err}$ for multiscale estimation over the range $v_x = v_y \in [-6, 6]$.

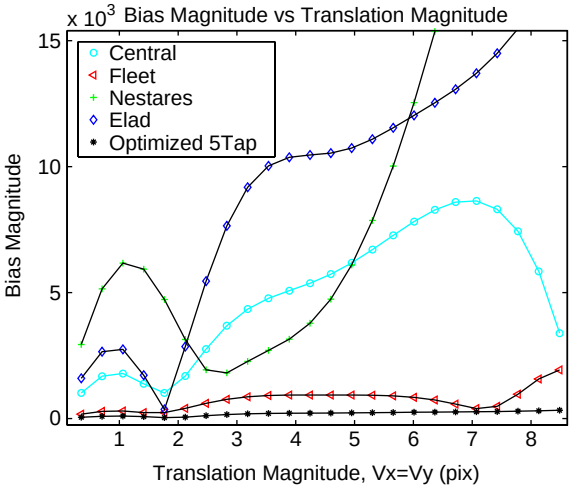

Fig. 12. Magnitude of registration error (bias) $\|\mathbf{b}(\mathbf{v})\|$ vs translation magnitude $\|\mathbf{v}\|$ where $v_x = v_y$ for the Tree image.

For the sake of completeness, we evaluate the performance of the optimized filters at different imaging SNR. As before, we perform MC simulations at each SNR to measure the MSE of the multiscale approach in estimating $\mathbf{v}$ along the line $v_x = v_y \in [0, 6]$, this time in increments of $\frac{1}{2}$ pixels. We use the same experimental setup as used before to produce Fig. 10, using a three level multiscale approach. Here, we see that the optimized filters outperform the other filters for SNR greater than about 25 dB. In fact, below this SNR, the performance of the optimized filters for multiscale estimation degrade substantially. In this SNR regime, the MSE is no longer dominated by

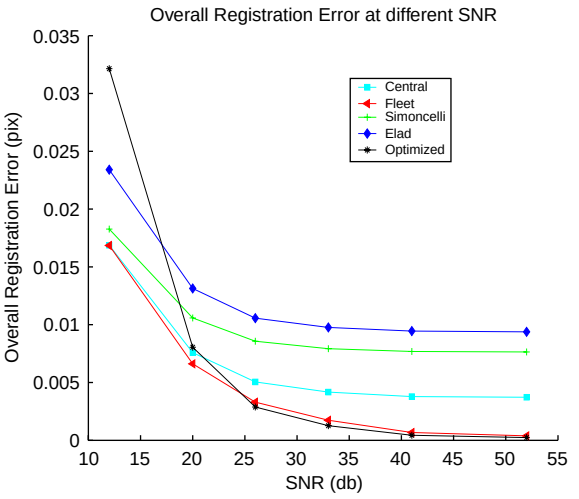

Fig. 13. Overall multiscale estimation error $\overline{Err}$ at different SNR over $v_x = v_y \in [0, 6]$ for the Tree image.

estimator bias. It is apparent that at such SNRs, minimizing bias is no longer a suitable objective for improving overall performance in the sense of MSE. We note, however, that SNR below 25 dB represents a very noisy scenario not often encountered in typical video imaging and rarely, if ever, addressed in the gradient-based motion estimation literature (Fig. 13).

4. Conclusions and future work

In this paper, we have presented and exploited the fundamental relationship between gradient-based motion estimator bias and the choice of gradient filters. We have proposed a cost function which captures the registration performance for a set of filters, and for a given image. We have shown an efficient mechanism for using this cost function to design bias-minimizing gradient filters for image registration. We have experimentally verified the utility of such optimized filters for improving estimator performance and suggested a means of incorporating the filter design process into a multiscale estimation framework, providing substantially improved estimation. Furthermore, we showed that the proposed filter design method can produce generic filters that offer performance improvements almost as significant as the image-

specific filters. Finally, we showed that the proposed optimized filters offer improved overall MSE performance at a wide range of imaging SNRs.

The work presented here suggests several possible directions for future work. For instance, one could study the problem of gradient-based image registration for higher order transformations such affine or projective. Another extension might examine the possibility of designing filters for dense optical flow field estimation where a different motion vector is estimated for every pixel in the image. Ideally, a practical algorithm would result, allowing for dynamic gradient filter coefficient adjustments based on local spectral properties in the image to optimize estimation performance. While the work of this paper has focussed on the high SNR situation, further investigation into the performance of such gradient-based estimators for low SNR scenarios would certainly prove beneficial for certain applications.

References

[1] J. Barron, D. Fleet, S. Beauchemin, T. Burkitt, Performance of optical flow techniques, CVPR 92 (1992) 236–242.

[2] J.W. Brandt, Analysis of bias in gradient-based optical flow estimation, Proc. IEEE Asilomar Conf. Signals Syst. Comput. (1995) 721–725.

[3] K.R. Castleman, Digital Image Processing, Prentice-Hall, Toronto, 1996.

[4] M. Elad, P. Teo, Y. Hel-Or, Optimal filters for gradient-based motion estimation, Proc. Int. Conf. Comput. Vision (September 1999) 559–565.

[5] M. Elad, P. Teo, Y. Hel-Or, On the design of optimal filters for gradient-based motion estimation, Int. J. Math. Imaging Vision, accepted for publication.

[6] S. Farsiu, D. Robinson, M. Elad, P. Milanfar, Fast and robust multi-frame superresolution, IEEE Trans. Image Process. 13 (10) (October 2004) 1327–1344.

[7] D.J. Field, Relations between the statistics of natural images and the response properties of cortical cells, J. Opt. Soc. Am. A 4 (12) (December 1987) 2379–2393.

[8] S.M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory, Prentice-Hall, 1993.

[9] J. Kearney, W. Thompson, D. Boley, Optical flow estimation: an error analysis of gradient-based methods with local optimization, IEEE Trans. Pattern Anal. Mach. Intel. 9 (2) (March 1987) 229–244.

[10] B. Porat, A Course in Digital Signal Processing, Wiley, New York, 1997.

[11] D. Robinson, P. Milanfar, Fundamental performance limits in image registration, IEEE Trans. Image Process. 13 (9) (September 2004) 1185–1199.

[12] E. Simoncelli, Design of multi-dimensional derivative filters, Proc. IEEE ICIP 1994 (1994).

[13] C. Stiller, J. Konrad, Estimating motion in image sequences, IEEE Signal Process. Mag. 16 (July 1999) 70–91.