

Gaussian Blurring-Invariant Comparison of Signals and Images

Zhengwu Zhang, Eric Klassen, and Anuj Srivastava, *Senior Member, IEEE*

Abstract—We present a Riemannian framework for analyzing signals and images in a manner that is invariant to their level of blurriness, under Gaussian blurring. Using a well known relation between Gaussian blurring and the heat equation, we establish an action of the blurring group on image space and define an orthogonal section of this action to represent and compare images at the same blur level. This comparison is based on geodesic distances on the section manifold which, in turn, are computed using a path-straightening algorithm. The actual implementations use coefficients of images under a truncated orthonormal basis and the blurring action corresponds to exponential decays of these coefficients. We demonstrate this framework using a number of experimental results, involving 1D signals and 2D images. As a specific application, we study the effect of blurring on the recognition performance when 2D facial images are used for recognizing people.

Index Terms—Gaussian blur, blur-invariant metric, Riemannian framework, geodesic distance, path straightening.

I. INTRODUCTION

IMAGES are often collected with a certain random amount of blurring present in them. There are several reasons for this blurring. The point spread function of an imaging device introduces some levels of blurring in the captured images. The techniques for compression, storage, transmission and processing can also result in blurring. Other effects such as motion, overexposure, strong illumination, etc, can also potentially introduce blur in images. Shown in Fig. 1 are two examples of images and their blurred versions. It has been one of the longstanding problems in image analysis to analyze, compare, and evaluate images while being invariant to their blurs. A common solution has been to *deblur* images using one of the many techniques available for deblurring and then analyze the deblurred images [1]–[8]. The problem with this approach is that here one has to estimate the amount of blurring [9], [10] in images and then try to deblur them, but the amount of blur has neither been easily defined nor easily estimated in the past. For example, the two blurred images in Fig. 1 have different levels of blurring but it is difficult to quantify

Manuscript received October 24, 2012; revised March 3, 2013 and April 12, 2013; accepted April 12, 2013. Date of publication April 24, 2013; date of current version June 4, 2013. This work was supported by the National Science Foundation under Grants NSF DMS 0915003, NSF DMS 1208959, and NSF 1217515. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sina Farsiu.

Z. Zhang and A. Srivastava are with the Department of Statistics, Florida State University, Tallahassee, FL 32304 USA (e-mail: zhengwu@stat.fsu.edu; anuj@stat.fsu.edu).

E. Klassen is with the Department of Mathematics, Florida State University, Tallahassee, FL 32304 USA (e-mail: klassen@math.fsu.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2013.2259840



Fig. 1. Examples of pairs of original (left) and blurred images (right).

that without knowing the original blur parameters. Therefore, in a deblurring approach, one is left with an unresolved issue of amount of deblurring required. A different approach is to extract relevant features that are actually invariant to blurring of images and use them for comparing images. Several authors have used algebraic moments to define such features [11]–[15].

Another challenge in this problem area is that the blurring kernel (or the point spread function) of an imaging system is typically unknown. There are several methods, termed blind deconvolution methods, that either estimate the kernel from the given data or by-pass the estimation process in some way. The feature-based methods mentioned above can be invariant to the nature or size of kernels. Taking a different approach, Gopalan *et al.* [16] provide a remarkable solution where they quotient out the variability associated with kernel as well as level of blurring. Another solution, albeit less general, is to assume that the blurring kernel belongs to a parametric family and estimate the parameter from data. The most common choice has been the Gaussian kernel and is broadly applicable in imaging scenarios [17], [18]. The question then becomes: What performance gain does this additional Gaussian assumption lead to, over the methods that do not assume any structure? This is the main focus of our paper. We show that, with a proper mathematical framework, one can exploit this Gaussian structure and provide improvements in analysis of blurred images.

A. Past Work

An interesting direction is to seek metrics and representations in *image spaces* (as opposed to feature spaces) that

allow analysis while being invariant to their levels of blur. This is motivated in part by techniques in shape analysis where objects are compared using metrics that are invariant to similarity transformations (translation, rotation, and scale). Zhang *et al.* [18] introduced this idea by formulating Gaussian blurring as an action of a one-parameter semigroup on the space of images. Using a log-Fourier representation of images, the orbits of this blurring semigroup are shown to be straight lines and the authors imposed Euclidean distances between orbits to compare given images. While this approach seems promising, in principle, it faces some practical issues – not only do logs of complex numbers represent computational issues but also small pixel values (close to zero) become large negative numbers in a log representation and disproportionately influence image comparisons. We seek a new mathematical representation and the corresponding metric that avoids these problems while continuing to be invariant to the blurring action.

Gopalan *et al.* [16] studied subspaces resulting from convolving an image with a complete set of orthonormal basis functions of a pre-specified maximum size. They showed that the subspaces created from an image and its blurred versions are identical. Then, they performed recognition and classification of blurred face images using the geometry of Grassmann manifolds. The strength of this method is that it does not assume a specific point spread function (PSF). However, it requires a pre-specification of the size of kernels. In case one underestimates this value, the results deteriorate fast and if one overestimates the size computational cost rises exponentially.

As mentioned above there are several papers on deriving blur-invariant features that can be used as surrogates for image comparisons. Flusser *et al.* [11], [12] introduced blur-invariant central moments, which are invariant to convolution of an image with an arbitrary symmetric kernel. Several others also proposed similar features [13]–[15], [19]. The advantages of these invariant moments are that they are easy to calculate and are invariant to a symmetric blurring kernel. Their main limitation is that they are at best partial representatives of images. They loose certain image information in the sense that one cannot, in general, reconstruct images from these features. The mapping from the image space to the feature space is many-to-one and the inverse mapping of the feature points is seldom made precise. Also, the higher-order moments are vulnerable to noise, so one usually restricts to lower moments in practice. The order of magnitude of different moment invariants are quite different, which makes it hard to use them together in a comprehensive metric. Some of these features cannot distinguish symmetric objects in images due to their invariance to the convolution with a symmetric PSF.

B. Our Geometric Approach

In this paper we use a well-known idea of implementing blurring as a solution of the heat equation [20]. The novel part here is to form a family of solutions as a blurring orbit of an image and to define a functional G that quantifies the

amount of blurring inherent in an image. The level set of G defines a *section* of the image space under the blurring action, i.e. denotes the set of all images at the same blurring level. Using a finite-dimensional representation of images, under a complete orthonormal basis, this section is found to be an ellipsoid where one can compute distances between points using geodesic lengths. Geodesics on high-dimensional ellipsoids are not available analytically and are computed using a numerical path-straightening approach. In this framework, one can blur or deblur given images so that they lie on the same section (ellipsoid) and then compare them using geodesic distances. These distances are, thus, invariant to the blur levels of given images. This framework is similar in approach to Riemannian methods for shape analysis of objects where metrics invariant to similarity transformations are used. Given any two objects, one “aligns” them using similarity transformations and then compares the representations of the aligned objects. The resulting metric is then independent of the rotation, translation, and scaling of the input objects.

There are several important differences in our approach from [18]. Firstly, the log-Fourier representation is replaced by a finite set of coefficients under an orthonormal basis, thus avoiding logarithms of complex numbers. Secondly, here the blurring group does not act by isometries as was the case in [18]. Instead, we simply choose an orthogonal section of the action (in fact we have a choice of which orthogonal section we work with) and we put a natural metric on this section. Thirdly, the representation in [18] led to a vector space under the Euclidean metric and, therefore, the geodesics were straight lines, but here the resulting section is an ellipsoid and requires some additional work for computing geodesic distance.

The rest of this paper is organized as follows. In Section II, we describe our framework that includes a finite-dimensional representation of images, formulation of blurring as a group action, a quantification of blurring using sections, and computation of geodesic distances on these sections. In Section III, we summarize the overall algorithm and present experimental results in Sections IV for 1D signals and 2D images. Section V provides a brief summary of the paper and Section VI concludes the paper with a discussion on potential statistical analysis of images under the proposed metric.

II. MATHEMATICAL FRAMEWORK

We start by describing the mathematical setup. Let D be the domain of signals or images of interest and let \mathcal{F} denote the set of smooth functions $D \rightarrow \mathbb{R}$. Also, let $L: \mathcal{F} \rightarrow \mathcal{F}$ denote the standard Laplacian operator on \mathcal{F} . The special cases that we will consider in this paper are:

- 1) In the case of 1D signals, $D = \mathbb{S}^1$ and $L \cdot f = -\frac{\partial^2 f}{\partial x^2}$.
- 2) In case of 2D images, $D = \mathbb{S}^1 \times \mathbb{S}^1$ and $L \cdot f = -(\frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2})$ for $x = (x_1, x_2)$.

Although Gaussian blurring of signals and images is commonly implemented as a convolution, it can also be modeled as a solution of the heat equation [20]. The heat equation governs the time evolution of a function $f: \mathbb{R} \times D \rightarrow \mathbb{R}$ using

the second-order spatial derivative according to:

$$\frac{\partial f(t, x)}{\partial t} = -(L \cdot f)(t, x). \quad (1)$$

If we set the initial condition to be a given function, say $f_0 \in \mathcal{F}$, i.e. $f(0, x) = f_0(x)$, then the solution $f(t, \cdot)$, also called the *forward solution*, denotes a blurred version of f_0 with the amount of blurring being controlled by t . As t increases, so does the blurring. It is interesting to note that the heat equation can also be solved in the negative time direction, called the *backward solution*, resulting in a *deblurring* of f_0 .

A. Finite-Dimensional Representations

In order to analyze these functions, we will use a finite representation, based on their coefficients under a complete orthonormal set. Since L is a self-adjoint positive definite operator, we know that \mathcal{F} (with the \mathbb{L}^2 metric) has a complete orthonormal Hilbert basis $\{\phi_0, \phi_1, \phi_2, \dots\}$, where each ϕ_n is an eigenfunction of L with eigenvalue λ_n , i.e. $L \cdot \phi_n = \lambda_n \phi_n$. Assuming that ϕ_0 is a constant function, we have that $\lambda_0 = 0$, and all other λ_n 's are greater than zero (due to positive definiteness of L). Any element $f_0 \in \mathcal{F}$ can then be expressed as $f_0(x) = \sum_{n=0}^{\infty} c_n \phi_n(x)$. In practice, since we are going to deal with finite data, we will use a basis set of size $N < \infty$ to make this representation finite. Each function $f_0 \in \mathcal{F}$ will be represented by a vector $c \equiv \{c_n, n = 0, \dots, N\} \in \mathbb{R}^{N+1}$. We will use Π to denote the projection from \mathcal{F} to \mathbb{R}^{N+1} : $\Pi(f_0) = c$. Clearly, Π is a many-to-one map and its inverse is set valued. However, we will use $\Pi^{-1}(c)$ to imply a specific element of this set given by $\sum_{n=0}^N c_n \phi_n(x)$, which denotes an approximation of $f_0(x)$.

If $f(t, x)$ is the solution of the heat equation, with $f_0(x) = \sum c_n \phi_n(x)$ as the initial condition, then what is the finite-dimensional representation of $f(t, \cdot)$? Using simple calculus, we have:

$$L \cdot f_0 = L \cdot \left(\sum_{n=1}^N c_n \phi_n \right) = \sum_{n=1}^N c_n (L \cdot \phi_n) = \sum_{n=1}^N c_n \lambda_n \phi_n.$$

If we set $f(t, x) = \sum_{n=0}^N e^{-\lambda_n t} c_n \phi_n(x)$, then we can verify that it satisfies the heat equation (Eqn. 1) since $L \cdot f(t, \cdot) = \sum_{n=0}^N e^{-\lambda_n t} c_n \lambda_n \phi_n$ which is exactly equal to $-\frac{\partial f(t, x)}{\partial t}$. In other words, $\tilde{c} \in \mathbb{R}^{N+1}$, with $\tilde{c}_n = e^{-\lambda_n t} c_n$, represents a blurred version of f_0 , with t controlling the level of blurring.

B. Blurring as a Group Action

Rephrasing this blurring in algebraic terms, we can form a blurring group \mathbb{R} that acts on the representation space \mathbb{R}^{N+1} according to the mapping:

$$\mathbb{R} \times \mathbb{R}^{N+1} \rightarrow \mathbb{R}^{N+1}, \quad (t, c) = \tilde{c}, \quad \tilde{c}_n = e^{-\lambda_n t} c_n. \quad (2)$$

We can easily check that this mapping is a group action (see [21] for a definition of group action). Thus, associated with each signal $f_0 \in \mathcal{F}$, and its finite representation $c \in \mathbb{R}^{N+1}$, is an orbit under the blurring group action:

$$[c] = \{\tilde{c} \in \mathbb{R}^{N+1} | \tilde{c}_n = e^{-\lambda_n t} c_n \forall n, \text{ for some } t \in \mathbb{R}\} \subset \mathbb{R}^{N+1}.$$

Any two functions f_1, f_2 with $\Pi(f_1) = c^{(1)}$ and $\Pi(f_2) = c^{(2)}$ belong to the same orbit if and only if one is a blurred or deblurred version of the other, i.e. $c^{(1)} \in [c^{(2)}]$ implies that $\exists t \in \mathbb{R}$, such that $\{c_n^{(1)} = e^{-\lambda_n t} c_n^{(2)} \forall n\}$.

C. Quantifying Blurring Using Sections

Given any two signals our goal is to compare them irrespective of their levels of blurring. Our strategy is to take arbitrary input signals, bring them to a fixed level of blurring, and then compare them using an appropriate metric. Thus, we need a way to quantify the level of blurring in a signal. This is accomplished using a *section* under the blurring action.

Definition 1: A subset S is an orthogonal section of \mathbb{R}^{N+1} under the action of the blurring group \mathbb{R} if: (i) one and only one element of every orbit $[c]$ in \mathcal{F} is present in S , and (ii) the set S is perpendicular to every orbit at the point of intersection.

The last property means that if S intersects an orbit $[c]$ at \tilde{c} , then $T_{\tilde{c}}(S) \perp T_{\tilde{c}}([c])$. Here $T_{\tilde{c}}(S)$ and $T_{\tilde{c}}([c])$ denote the spaces tangent to the sets S and $[c]$, respectively, at the point $\tilde{c} \in [c]$. For the action specified by Eqn. 2, we can construct an orthogonal section S as follows. First, define a functional $G: \mathcal{F} \rightarrow \mathbb{R}$ by $G(f_0) = \int_D f_0(x) (L \cdot f_0)(x) dx$. Using integration by parts, we can also write G as $G(f_0) = \int_D \langle \nabla f_0(x), \nabla f_0(x) \rangle dx$. Note that this is true because the domain D does not have any boundary. Since we will represent functions by their coefficients, it is convenient to express G directly in terms of the coefficient as $\tilde{G}: \mathbb{R}^{N+1} \rightarrow \mathbb{R}$ given by $\tilde{G}(c) = G(\Pi^{-1}(c)) = \sum_{n=0}^N \lambda_n (c_n)^2$. As mentioned earlier, $\lambda_0 = 0$ in the case ϕ_0 is a constant (which is the case in this paper). Thus, the summation can start from $n = 1$ instead of $n = 0$.

Now, for a positive real constant $\kappa > 0$, we define a section S_{κ} of \mathbb{R}^N under the group action of \mathbb{R} as $S_{\kappa} = \tilde{G}^{-1}(\kappa)$. Since $S_{\kappa} = \{c \in \mathbb{R}^N | \sum_{n=1}^N \lambda_n (c_n)^2 = \kappa, \kappa > 0\}$, and since all the λ_n 's are positive, S_{κ} is actually an $(N - 1)$ -dimensional **ellipsoid** in \mathbb{R}^N . Note that this definition does not constrain the value of c_0 which is left arbitrary. One way to handle that is to restrict to all the functions that satisfy $\int_D f(x) dx = 1$; this makes $c_0 = 1 / (\int_D \phi_0(x) dx)$, a constant. Since the value of function G does not depend on c_0 , one can just leave it as a constant. At any point $c \in S_{\kappa}$, the space normal to S_{κ} (inside \mathbb{R}^N) is a one-dimensional space spanned by the vector $\mathbf{n}_c = \{\lambda_1 c_1, \lambda_2 c_2, \dots, \lambda_N c_N\}$. Let \mathbf{u}_c denote the unit vector in the normal direction $\mathbf{u}_c = \mathbf{n}_c / \|\mathbf{n}_c\|$. Therefore, the tangent space $T_c(S_{\kappa}) = \{v \in \mathbb{R}^N | v \perp \mathbf{u}_c\} \subset \mathbb{R}^N$. We endow S_{κ} with the Euclidean metric: for any $v_1, v_2 \in T_c(S_{\kappa})$, $\langle v_1, v_2 \rangle$ is the standard Euclidean inner product in \mathbb{R}^N . Also, note that for any $\tilde{c} \in [c]$, $T_{\tilde{c}}([c])$ is a one-dimensional space spanned by \mathbf{u}_c .

Lemma 1: S_{κ} , as defined above, is an orthogonal section of \mathbb{R}^N under the action of the blurring group.

Proof: We need to verify the two properties listed in Definition 1. (1) The function $t \mapsto \sum_n e^{-2\lambda_n t} \lambda_n c_n^2$ is a continuous, strictly monotonically-decreasing function that ranges from $+\infty$ to 0. Thus, for any $c \in \mathbb{R}^N$ and $\kappa > 0$, there exists a unique t^* such that $\sum_n e^{-2\lambda_n t^*} \lambda_n c_n^2 = \kappa$. (2) Note that $T_c([c])$ is spanned by \mathbf{u}_c , which is also the direction of gradient of \tilde{G} . Since S_{κ} is a level set of \tilde{G} , it is automatically perpendicular to

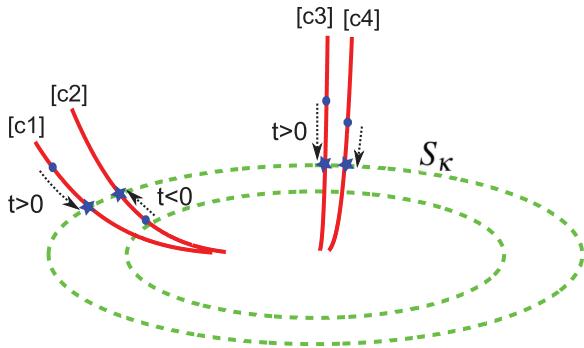


Fig. 2. Pictorial depiction of the orbits and the orthogonal sections S_K .

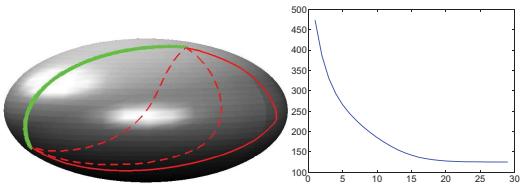


Fig. 3. Examples of path straightening on an ellipsoid in \mathbb{R}^3 : the red solid lines show the initial path, the red broken lines are intermediate results of the path straightening, and the green solid lines show the final geodesics. The energy plots show a decrease in E versus path-straightening iterations.

\mathbf{u}_c and $T_c([c])$. In other words, the orbits are just the flow lines for the gradient vector field of the function \tilde{G} and since the level sets of a function are perpendicular to the flow lines of gradient of that function, it follows that the S_κ is perpendicular to these orbits. ■

How does one blur or deblur signals in this scheme? Let f_0 be a given signal with representation $c \in \mathbb{R}^N$ such that $\tilde{G}(c) = \kappa_1$. Let's say we want to change its blurring level to another value κ_2 ($\neq \kappa_1$). That is, we need to find a $t^* \in \mathbb{R}$ such that:

$$\sum_{n=0}^N \lambda_n e^{-2\lambda_n t^*} c_n^2 = \kappa_2, \quad \text{where} \quad \sum_{n=0}^N \lambda_n c_n^2 = \kappa_1. \quad (3)$$

This is akin to finding the intersection of the orbit $[c]$ with the level set S_{κ_2} . Due to the monotonicity and continuity of the leftmost term in t , we can use the bisection method to solve for t^* . Fig. 2 shows a pictorial illustration of the orbits and orthogonal sections.

The next issue is to quantify difference in images that are at the same blurring level, i.e. their representations in \mathbb{R}^N fall on the same section S_κ for some κ . Since S_κ is an ellipsoid in \mathbb{R}^N , we can quantify image differences using the geodesic length between their representations in \mathbb{R}^N . Although it is possible to have analytical expressions for geodesics on ellipsoids in smaller dimensions, these formulas get very complicated as the dimension grows. Appendix A describes a numerical technique, called *path-straightening* introduced in Klassen *et al.* [22], for computing geodesics between points in S_κ . We will define the length of this geodesic path as d_κ to define a blur-invariant metric between images. Shown in Fig. 3 is an example of geodesics computed using this algorithm on an ellipsoid in \mathbb{R}^3 .

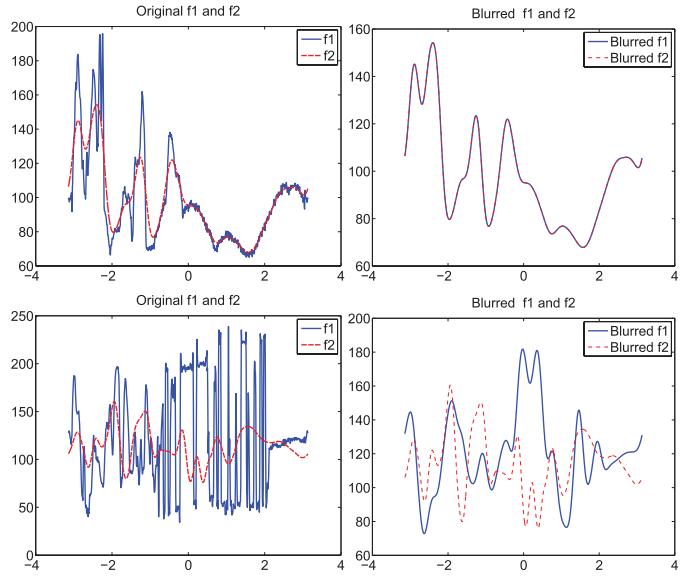


Fig. 4. Top row shows an artificial case where f_2 is a blurred version of f_1 . Bottom row shows two different signals at different blurring levels. In each row, original f_1 and f_2 are shown in the left, and their representations at the same blurring level are shown in the right.

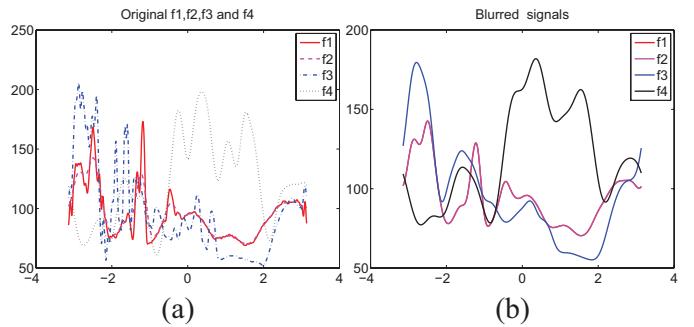


Fig. 5. Comparing several signals together. (a) shows original signals, and (b) shows signals blurred to the same blurring level.

III. OVERALL ALGORITHM FOR COMPARING SIGNALS

Now we are ready to outline the procedure for comparing signals at given arbitrary blurring level. Given any two functions, f_1 and f_2 , our goal is to first bring them to the same blurring level, say κ , and then find the geodesic distance between them in S_κ .

Select a number N for truncating the basis expansion and a $\kappa > 0$. Typically κ is the smallest G -value of the given signals. Select a truncated orthonormal basis set $\{\phi_0, \phi_1, \dots, \phi_N\}$ with the corresponding eigenvalues $\lambda_0, \lambda_1, \dots, \lambda_N$ for the Laplacian L .

- 1) Evaluate the discrete representations of f_1 and f_2 using the projection $c^{(j)} = \Pi(f_j)$, $j = 1, 2$.
- 2) Using the bisection method and Eqn. 3, find values of t_1 and t_2 so that the vectors $\{e^{-t_1 \lambda_n} c_n^{(1)}\}$ and $\{e^{-t_2 \lambda_n} c_n^{(2)}\}$ are both in S_κ .
- 3) Find the geodesic α between these two points in the ellipsoid S_κ using the path straightening algorithm given in the Appendix A.
- 4) Compute the length of this path using $\sum_t \|\frac{d\alpha}{dt}(\tau)\|$. Set $d_\kappa(f_1, f_2)$ to this value.

Next, we illustrate this algorithm using examples of 1D signals and 2D images.

IV. EXPERIMENTAL RESULTS

In this section, we will show some experimental results of the proposed framework. Four types of experiments are presented: (1) comparisons of 1D signals (2) comparisons of 2D images (3) image classification with blurring-invariant metric d_κ , and (4) face recognition using blurred 2D images.

A. Analysis of 1D Signals

For the 1D periodic signals, we set $D = \mathbb{S}^1$ and use the orthonormal basis: $\{\frac{1}{\sqrt{2\pi}}, \frac{\cos x}{\sqrt{\pi}}, \frac{\sin x}{\sqrt{\pi}}, \frac{\cos 2x}{\sqrt{\pi}}, \dots, \frac{\sin mx}{\sqrt{\pi}}, \frac{\cos mx}{\sqrt{\pi}}\}$. The Laplacian operator is given by $L = \frac{\partial^2}{\partial x^2}$, so that $L \cdot \phi_n = \lambda_n \phi_n$, where $\lambda_n = n^2$. In the experiments described below, we will use $N = 2m + 1 = 51$ to analyze signals sampled at 200 points along their domain \mathbb{S}^1 . First, we take a signal say f_1 and artificially generate f_2 by blurring f_1 (using the heat equation for a certain parameter t). Thus, $G(f_2) < G(f_1)$ and we set $\kappa = G(f_2)$. Now, using Eqn. 3 we can find t^* , the level of blur of f_1 that brings it to the blur level of κ . Fig. 4 (top row) shows an example where f_1, f_2 are shown in the left and their representations at the same blur level are shown in the right. In the second row we compare two different signals at different original blurring levels and repeat the earlier process.

Next, we compare four signals simultaneously, as shown in Fig. 5 (a). These four signals are at very different blurring levels. Fig. 5 (b) shows the four signals brought to the same blurring level (i.e. in the same S_κ) and one can verify visually that all the signals are at the same blur level.

B. Analysis of 2D Images

In this case $D = \mathbb{S}^1 \times \mathbb{S}^1$ or \mathbb{T}^2 , the two dimensional torus and L denotes the Laplace operator $L = -\frac{\partial^2}{\partial x_1^2} - \frac{\partial^2}{\partial x_2^2}$.

To define an orthonormal basis for \mathbb{L}^2 functions on \mathbb{T}^2 , we will simply take the basis in \mathbb{S}^1 and form cross products. In x_1 direction, we use $\{\frac{1}{\sqrt{2\pi}}, \frac{\cos x_1}{\sqrt{\pi}}, \frac{\sin x_1}{\sqrt{\pi}}, \dots, \frac{\sin m_1 x_1}{\sqrt{\pi}}, \frac{\cos m_1 x_1}{\sqrt{\pi}}\}$. Similarly, in the x_2 coordinate, the basis elements are $\{\frac{1}{\sqrt{2\pi}}, \frac{\cos x_2}{\sqrt{\pi}}, \frac{\sin x_2}{\sqrt{\pi}}, \dots, \frac{\sin m_2 x_2}{\sqrt{\pi}}, \frac{\cos m_2 x_2}{\sqrt{\pi}}\}$. With the combination of these two orthonormal bases, we have basis for 2D images $\phi_n(x_1, x_2), n = 1, 2, \dots, N, N = (2m_1 + 1)(2m_2 + 1)$. The eigenvalue corresponding to the basis element $\frac{\cos(n_1 x_1) \sin(n_2 x_2)}{\pi}$ is $n_1^2 + n_2^2$. To evaluate d_κ for size $n \times n$ images, we will use $(n + 1)^2$ basis elements.

In the following experiments, we will evaluate our method on images that have Gaussian blur, optics blur, and motion blur. We will use images with both similar and different contents, and a variety of blur levels to compute invariant metric and geodesic distances.

I. Gaussian Blurred Images: In this experiment we use the LIVE Image Quality Assessment Database [23] and CSIQ image database [24]; they include images that have been distorted using Gaussian blur at several different levels and

then resampled at size 200×200 . We will use $N = (201)^2$ basis elements to represent them exactly in our representation system.

At first we take images with the same content but different blur levels, and apply our algorithm to bring them to the same section S_κ . The common level κ is chosen to be the smaller of the two G -values. Figs. 6 and 7 show some results from this experiment but on images from different datasets. In each row the first two panels show the given images f_1 and f_2 , and the third column shows the difference image $|f_1 - f_2|$. After the image with higher G -value is blurred to bring it at the same blur level as the other, the result is shown in column (d). The new difference image is shown in column (e). In column (c) and (e) under each row, we also show the Frobenius norm of before and after difference images, denoted by F_{bf} and F_{af} respectively. On the right, we display the optimal blur parameter t^* , the percentage reduction in the Frobenius norm after blurring, the blur level κ , and the geodesic distance d_κ between images in S_κ . We can see that when the difference was initially not large, e.g. row 1 in Fig. 6, we can almost exactly match them by bringing them to the same blurring level, but while there is a small residual in case the original blurring difference was initially large (second row). In Fig. 7 we obtain large ($> 70\%$) reductions in the difference norm due to the blurring alignment.

A similar but more exhaustive experiment was performed using the CSIQ dataset and some summary results are shown in Fig. 8. This experiment uses 28 different images at five different blurred levels each. And 112 in-class pairwise comparisons were performed. The left panel shows the histogram of difference norm before (blue) and after (yellow), and the right panel shows the histogram of the percentage reductions in the Frobenius norm due to the blurring alignment. These results point to a large reduction (on average $\sim 65\%$) in difference norm due to matching of blur levels.

Finally, we show some results on matching images using real, astronomical images in Fig. 9. As shown, we are able to reduce the difference norm substantially using matching.

II. Optics Blurred Images: Next, we apply our method to optically blurred images taken from a Cannon EOS 10D camera. In collecting these pictures we fixed the focus and set the shutter speed to be automatic, but used different apertures. Since the aperture controls the amount of light reaching the image sensor, different apertures imply different exposure levels. The size of the aperture also governs the *depth of field* which measures the portion of the picture, from foreground to background, that is in sharp focus. A smaller aperture implies a greater depth of field and a larger aperture implies a more restricted depth of field. In Fig. 10, the first two columns show examples of these pictures. Thus, using different apertures and auto shutter speed, we introduce blur to the pictures. Using the same experiment as in Fig. 6 example, we get the results shown in Fig. 10. From these results we can see that the algorithm is able to moderately reduce ($\sim 17\text{--}34\%$) the difference norm between image pairs. This can be attributed to the fact that optics blur can only be roughly approximated by a Gaussian blur.

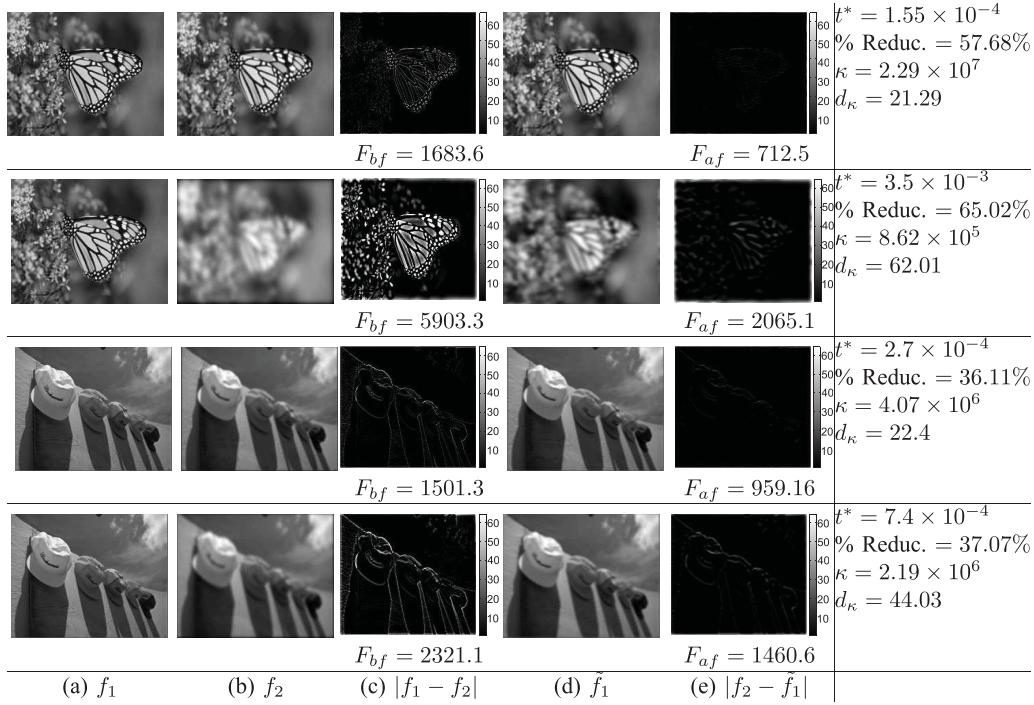


Fig. 6. (a) and (b) are two given images, f_1 and f_2 respectively, at different blur levels. We intentionally put the more blurred image in the second column. (c) is difference image between f_1 and f_2 . (d) is the blurred version of f_1 , denoted as \tilde{f}_1 , which is in the same S_K as f_2 . (e) is difference image between f_1 and \tilde{f}_2 (difference after matching their blur levels). Below (c) and (e) are the Frobenius norm of the difference images. In the right we show some quantitative details.

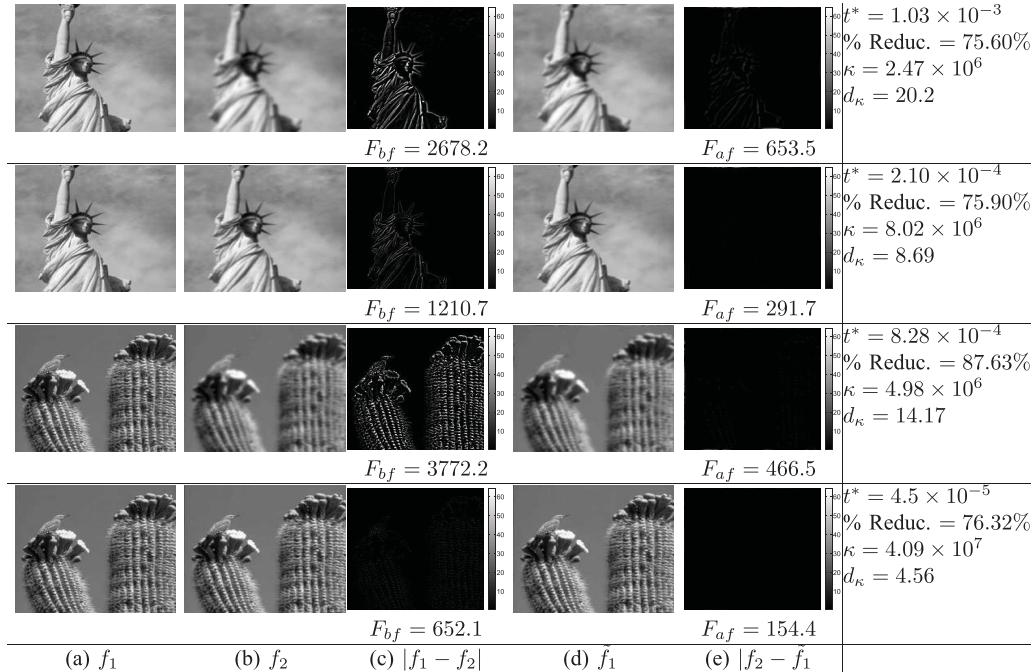


Fig. 7. Same as in Fig. 6 but involving Gaussian-blurred images in CSIQ dataset.

III. Motion Blurred Images: We have also applied this method to motion-blurred images even though this distortion is quite different from Gaussian blurring. Here the motion blurred images were simulated using Adobe Photoshop, and Fig. 11 shows some results. In Fig. 11, the first row is based on the vertical motion blur while the second row uses the horizontal motion blur. Since a Gaussian blurring is isotropic

and motion blurring is directional, one does not expect our framework to estimate the motion blur precisely. However, as shown in these examples, the framework can still quantify the blur level of images quite nicely. The images f_1 and f_2 in the last row has the same level of blur but in different directions (horizontal and vertical). Their G values are found to be similar and the images essentially remain unchanged

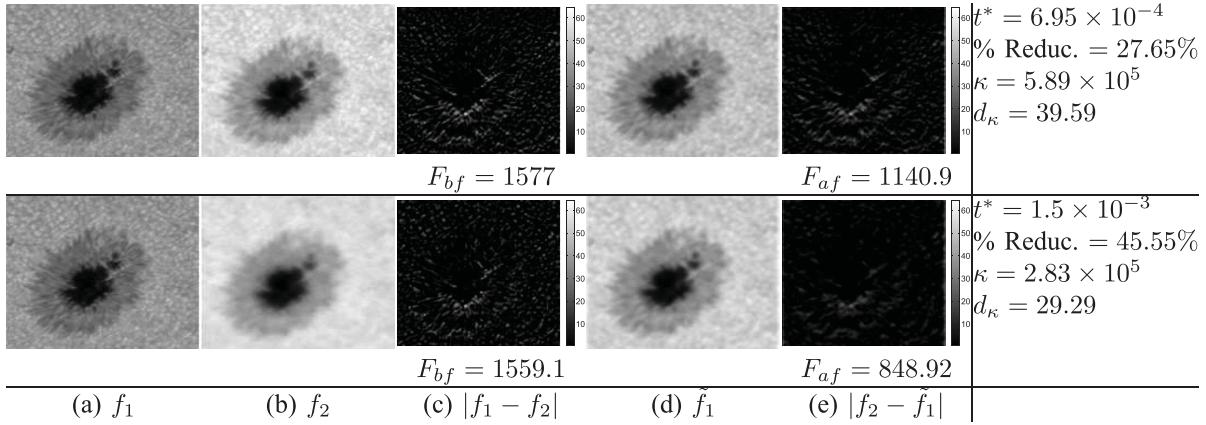


Fig. 9. Same as Fig. 6 using real astronomical images.

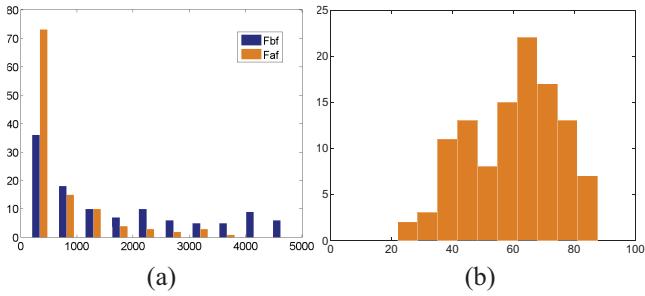


Fig. 8. An Exhaustive experiment for CSIQ dataset. (a) Histograms of the difference norm before matching (blue) and after (yellow); (b) Histogram of the percentage reduction in the difference norm.

during matching causing a small change in the difference norm.

Next we study images with different contents and at different blurring levels, and repeat the previous experiment with some results shown in Fig. 12. The first and second columns show images from the LIVE database, with more blurred images in the second column. Column (c) is the blurred version of (a) after matching it to the (b). In order to better illustrate the matching, we show some randomly picked rows from these images in (d), where it is easier to see that (b) and (c) now are at similar roughness levels.

Another advantage of this framework is that not only it can bring the less Gaussian blurred image to a higher blurred level, it also can deblur the image by using a negative value for t in Eqn. 3. However, in practice, the deblurring is feasible for only smaller values of $|t|$. The reason is that if a certain coefficient c_n is already zero, it cannot be scaled back. Fig. 13 shows a few examples of deblurring where we take two images with the same content but different blurring levels from the LIVE database. Instead of blurring one with larger G value, we deblur the one with the smaller G value to match it with the other.

C. Image Classification With Blurring Invariant Metric

Here we study the problem of image retrieval/classification using the proposed blurring-invariant metric d_κ . We use two datasets in this section: the ETH-80 [25] dataset and the ALOI dataset [26], [27].

TABLE I
% ERROR RATES FOR NN-CLASSIFICATION

Categories	$d_{\kappa_1} (S_{\kappa_1})$	\mathbb{L}^2	$d_{\kappa_2} (S_{\kappa_2})$	Blurred \mathbb{L}^2
Apple	0	0	12.20	63.41
Car	0	46.34	2.44	100.00
Cow	19.51	56.10	17.07	97.56
Cup	0	4.80	7.32	100.00
Dog	14.63	29.27	14.63	80.49
Horse	2.44	2.44	12.20	0
Pear	0	0	0	2.44
Tomato	0	0	0	0
Average Error Rate(%)	4.57	17.38	8.23	55.49

First we present results on a subset of ETH-80. Images in this dataset are cropped so that they contain only the object without any background, with the image resulting size being 128×128 pixels. The dataset contains 8 categories of different objects: apple, car, cow, cup, dog, horse, pear and tomato; each category has ten different objects and each object has 41 images taken from different imaging angles. We restrict to only one object from each category so that in total we have $8 \times 41 = 328$ images. Fig. 14 shows the 8 objects used in the experiment.

Table I presents the nearest-neighbor (NN) classification results using d_κ and the Euclidean metric, under two different values of κ . These rates are computed as follows. For each of the 328 images in the dataset, we first calculate its G -value, and choose the smallest G as the $\kappa_1 = 1.344 \times 10^6$, and blur all images to make them elements of S_{κ_1} . Then, taking each image as a query image, we perform the nearest neighbor classification under d_{κ_1} . The percentage of incorrect retrieval for each category in the dataset are shown in the second column. The third column's results are calculated using the nearest neighbor classifier under the Euclidean norm between images. To study the effect of κ on classification, we also choose a smaller $\kappa_2 = 2.767 \times 10^4$, and perform the nearest neighbor classification in S_{κ_2} ; this result is shown in fourth column. The last column result is obtained by blurring the query image to S_{κ_2} and using the Euclidean norm to find the nearest neighbor. We find that not only is d_κ superior

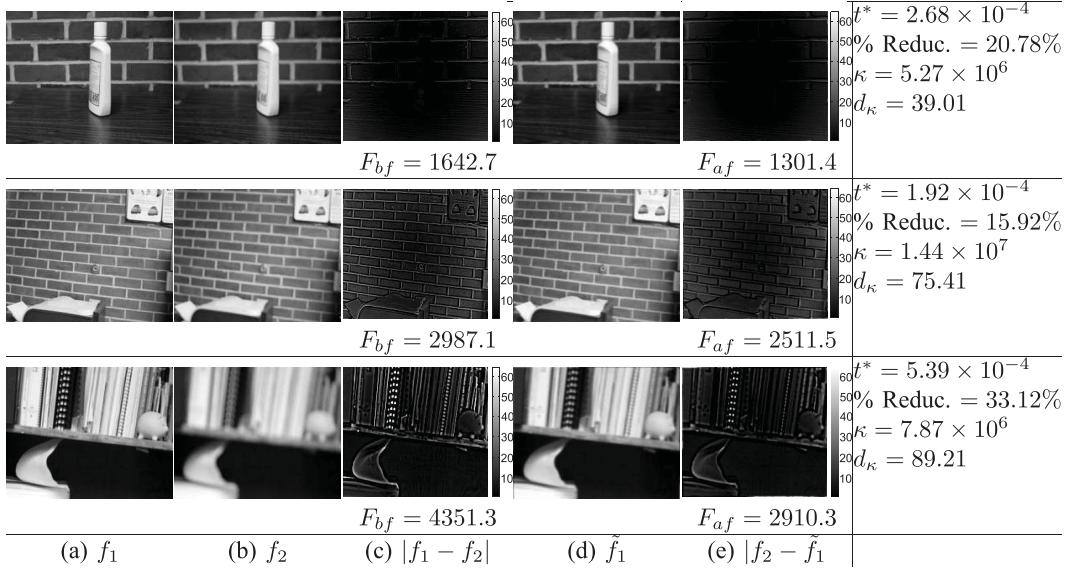


Fig. 10. Examples using images involving different camera apertures.

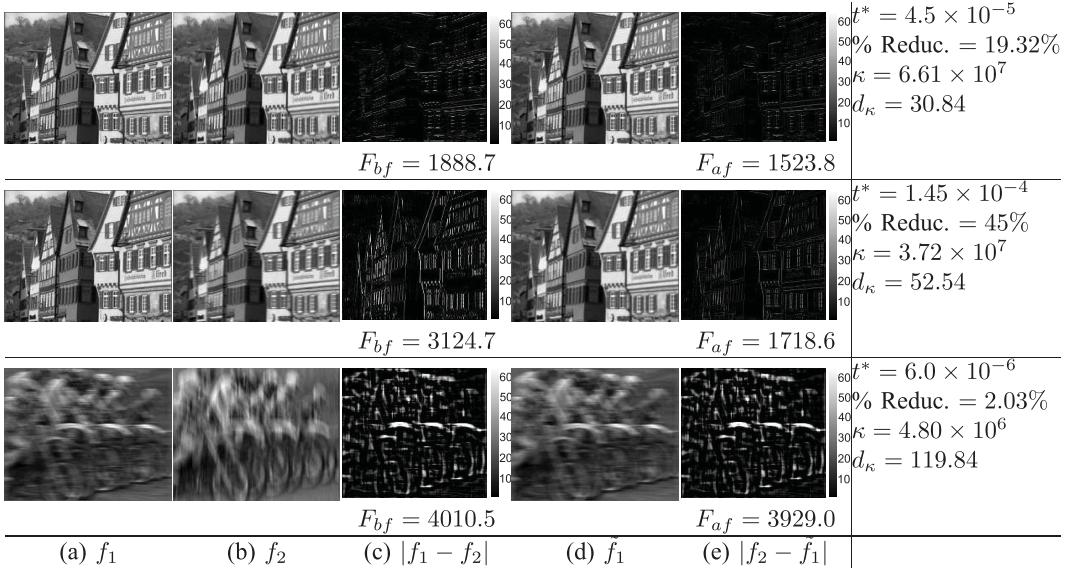


Fig. 11. Motion-blurred images. The first row is vertical motion blur. The second row is horizontal motion blur. The last row is horizontal blur versus vertical blur with the same kernel size.

to the Euclidean metric for classification, its performance is relatively stable with respect to a change in κ .

In order to study the classification performance with respect to the blur level of images, we perform a larger experiment as follows. For each query image we blur it to a certain level denoted by t and then compare the blurred query image with the other 327 images in the dataset using different distances being studied here. The classification rates for different blur levels are shown in Fig. 15(a) where the horizontal axis represents the blur level of query images. We clarify that we choose and fix a κ for the whole experiment. As the plot shows, the classification rate under d_κ , denoted by BI Metric, remains almost constant despite a very large increase in the blur levels of the query images. We compare our results with a variety of other representations/metrics: (1) The classification

performance for the log-Fourier transform metric [18], denoted by LFT Metric, is shown using the yellow dashed line in the plot. We can see that at lower blur levels this metric is stable but the performance falls steeply as the blur level gets high. The reason for this is that when the blurring level is high, the marginal Fourier transforms of images start being singular and their log-Fourier representations are matrices with large negative numbers. (2) We have implemented the *blind deconvolution* method presented in [28] for deburring the query images and then using the \mathbb{L}^2 distances between them and gallery images. The result is plotted with black solid line and denoted by BD. (3) From the approach presented in [11], denoted by CMI, we computed 54 central-moment invariants, upto moments of 13th order, and normalized each invariant with the maximum absolute value of that invariant. Then,

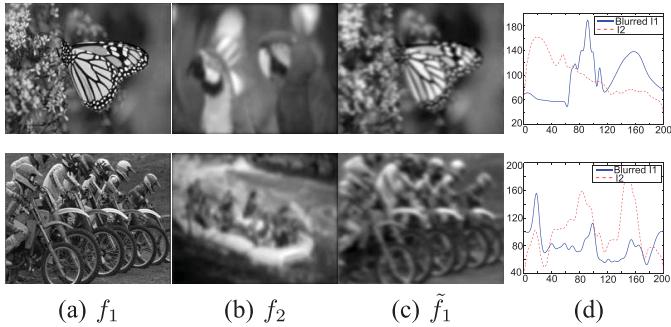


Fig. 12. Each row shows two images, f_1 and f_2 , at different blurring levels. f_1 is less blurred than f_2 . (c) is the blurred version of f_1 , denoted as \tilde{f}_1 , which is at the same blurring level with f_2 . (d) shows some selected rows from f_2 and \tilde{f}_1 .

we calculated the Euclidean distance between the coefficients corresponding to different images for classification. The result is plotted with green dotted solid line. (4) The figure also shows classification under the standard Euclidean metric, denoted by “Euclid Metric” in the plot. As the figure shows, our method provides the highest and most stable performance for classification of images under a large range of blur.

To study this comparison in a different scenario, we perform a similar classification experiment on ALOI dataset. ALOI is a color image collection of one thousand small objects with systematically varied viewing angle, illumination angle and illumination color for each object. We randomly chose 38 objects from this collection and selected the first 14 views of these objects to reduce the amount of images in the dataset. The 1NN classification results for different metrics are shown in Fig. 15(b). We can see that at the lower blur, all the different representations/metrics have almost the same classification rate. However, as the blur level increases, the classification rates of all other methods decrease steadily: faster for the Euclidean metric and the log-Fourier transform metric than the CMI representation. The rate of our blurring-invariant metric stays the same.

D. Face Recognition Using Blurred Images

An interesting application of a blur-invariant metric is in face recognition from 2D blurred images, as described in [16]. Similar to that paper, we will demonstrate the effectiveness of our metric for face recognition using synthetically blurred images taken from the extended YaleB dataset [29]. In this experiment we will compare five metrics: the blur-invariant metric d_κ proposed here, denoted by BI Metric; the blur-invariant metric based on the log-Fourier transform [18], denoted as earlier by LFT Metric; the blur-robust descriptor used in [16], denoted by BRD; the baseline Euclidean metric, denoted by Euclid Metric and the central moment-invariants [11], denoted by CMI. The YaleB dataset has 38 subjects under 64 illumination settings. We select the first 10 subjects and 10 illuminations from each subject for this recognition experiment, with the image size of 168×168 . Since for this size the BRD algorithm is too slow, it is left out of this particular experiment and the remaining four metrics are evaluated. Fig. 16(a) shows the 1NN face-recognition

TABLE II
COMPUTATIONAL TIME COMPARISON IN SECONDS

Imsize	LogFT	Ours (d_κ)	CMI(54)	BRD (Imsize/4)	BRD (Imsize/2)
32×32	0.0035	0.0806	0.3170	0.0216	0.1540
64×64	0.0059	0.2574	0.6067	0.4254	4.8782
128×128	0.1930	0.8347	2.0347	15.465	233.65

performance as the test images are increasingly blurred, over a large range of blur values. As can be seen in this plot, the performance resulting from d_κ holds constant over a large range of blurring. The LFT metric’s performance is stable for lower values of the blur parameter but it completely breaks down at higher values. The baseline Euclidean metric shows a steady decrease in the performance as the level of blur increases. CMI approach does not perform well here.

In order to include BRD in the experiment, we reduce the image size to 32×32 to help speedup the experiment. For implementing BRD, one needs to specify the upper limit on blur kernel size and we choose several values: 8, 16, 24 for this limit. Fig. 16 shows the recognition for different methods under a large range of blur parameter: [0, 0.2]. We found that for BRD, the blur kernel of size 16 provides the best recognition rate. Both d_κ and LFT metric provide nearly constant recognition rate despite a significant change in the blur level of test images. In contrast to the previous experiment, the LFT metric recognition rate doesn’t drop here because of the smaller size of images. In smaller images, the Fourier transforms will have only the lower frequencies and there is no problem in taking their logs. The BRD method performs well for small blurs but its recognition rate drops steadily as blur increases. Since the images of human faces are inherently symmetric, the CMI approach is found to perform poorly here. To evaluate these performance rates further, we zoom into a smaller range of [0, 0.05] and look at the recognition rates in Fig. 17. We can see that in the range 0–0.01 the BRD rates are quite stable for different values of the kernel size. However, the performance falls off as the blur parameter goes beyond 0.01. This experiment helps to establish the pros and cons of the BRD approach. Since this method does not assume any kernel structure, it is far more general than our method. However, the additional information of the Gaussian structure (without the knowledge of the blur parameter–standard deviation) is exploited well by our method to provide a superior performance even in presence of drastic blurring of images.

Table II shows the computational time (seconds) for each algorithm. These numbers are generated on a laptop with i5-2450M CPU and 8 GB RAM. A significant disadvantage of the BRD algorithm is its exponential computational cost as the image size and kernel width grows.

V. SUMMARY

We have presented a framework for quantification of blurring in signals and images, and for comparing them under the metric that are invariant to their levels of blur. The framework is based on forming a group action of the blurring group

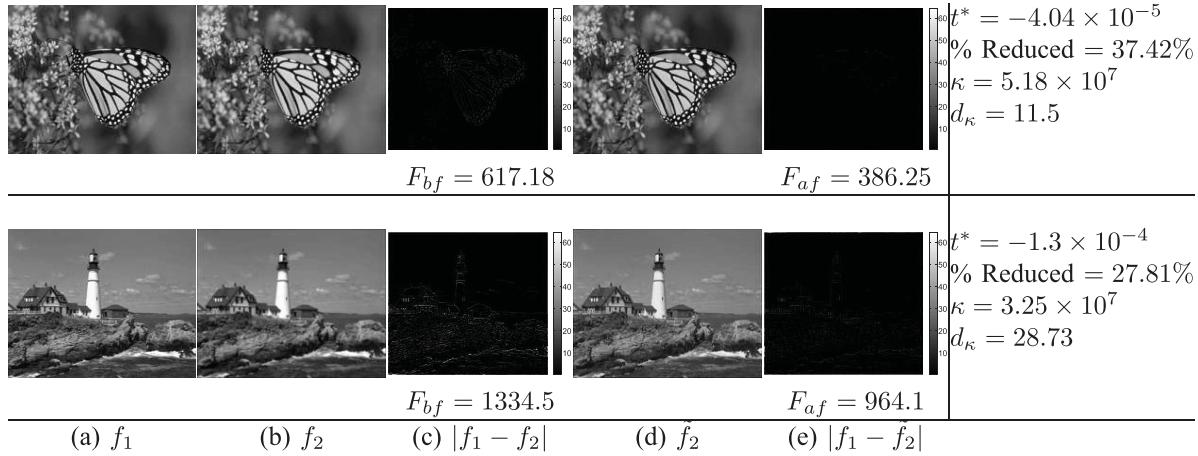


Fig. 13. Example of deblurring images. (a) and (b) are two original images, where (b) is more blurred than (a). (c) is the difference between image (a) and (b). (d) is the deblurred version of (b). (e) is the difference between (a) and (d).

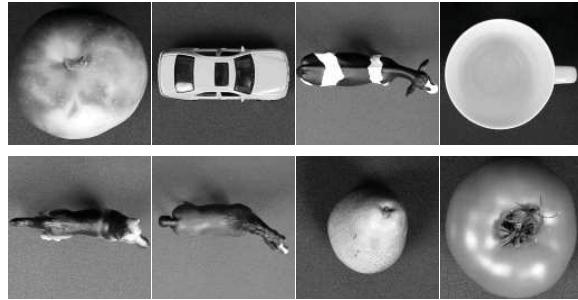


Fig. 14. Objects from ETH-80 database used in the classification experiment.

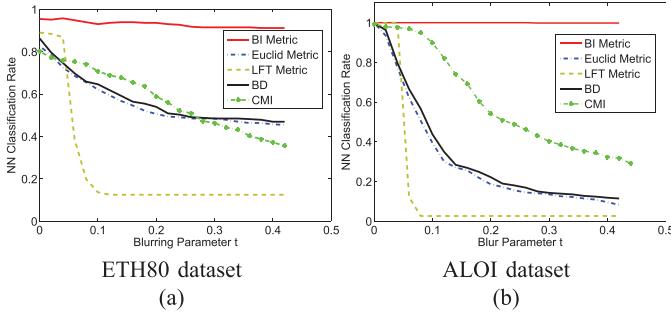


Fig. 15. ETH-80 and ALOI dataset nearest-neighbor (a) ETH-80 and (b) ALOI dataset nearest-neighbor classification result.

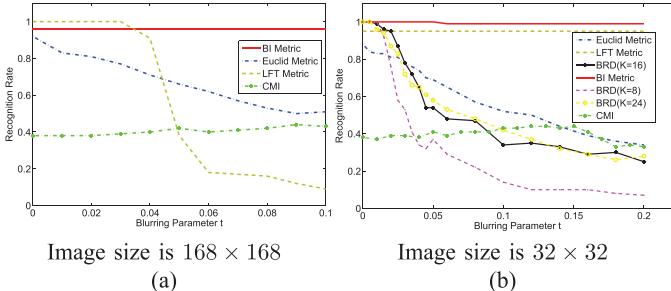


Fig. 16. Blurred face recognition comparison on YaleB subset. (a) image size is 168×168 , (b) image size is 32×32 .

on the signal and image spaces using solutions of the heat equation (which implements Gaussian blurring). These solu-

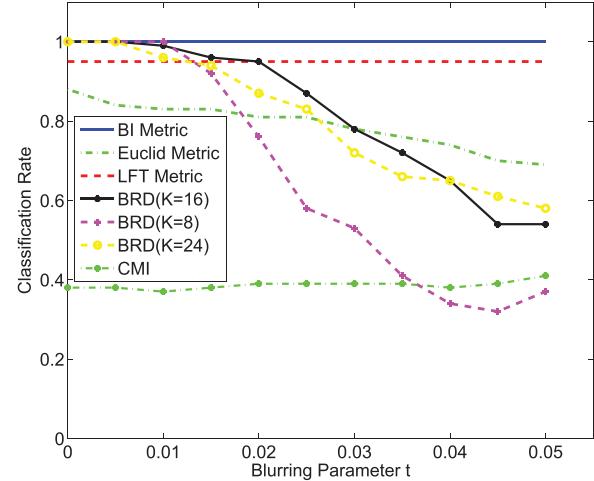


Fig. 17. Blurred face recognition comparison on YaleB subset (image size equals 32×32).

tions establish blurring orbits of images (an orbit denotes all possible blurrings of a signal or an image) and then we define a section set of this group action as the manifold where images are compared. This section set is chosen because it intersects each orbit only once and these intersections are orthogonal. In this set up any two given images are blurred/deblurred appropriately to find intersections of their orbits with the section, and one finds geodesic distances between these intersecting points on the section. In other words, the images are brought to the same level of blur and then compared using a proper metric; the metric is invariant to the level of blur chosen for comparison. The geodesic calculation is performed using a path straightening algorithm on ellipsoid formed by the section. Experimental results involving LIVE database, CISQ database, astronomical images, optically blurred images and motion blurred images show success of this framework in achieving the stated goals. The classification and blurred face recognition results show the advantage of our method compare with Euclidean metric, log-Fourier transform blurring invariant metric [18], blur-robust descriptor [16] and CMI [11].

VI. DISCUSSION

An important strength of our framework is that it provides geodesics and not just metrics for comparing image representations on S_κ . These geodesics in turn allow us to determine sample statistics for a set of given images; these statistics are based on d_κ and thus depend only on κ and not on the blur level of the given images. Taking the simplest statistic—sample mean—we will show how to obtain such a mean and its strengths relative to the Euclidean mean. Note that moment-based feature representations are not invertible, so an average in the feature space cannot be mapped back to the image space. Therefore, only the Euclidean case is used for comparisons. We focus on two scenarios:

I. Mean of images from the same orbit: Let f_0 be an image with $f_0(x) = \sum_{n=0}^N c_n \phi_n(x)$ and $\sum_{n=0}^N e^{-\lambda_n t} c_n \phi_n(x)$ be its Gaussian blurred version for $t \in \mathbb{R}_+$. Take k different blur versions of f_0 : $f^0(x), f^1(x), f^2(x), \dots, f^k(x)$ at the blurring parameter $t = t_0, t_1, t_2, \dots, t_k$. The average image under the Euclidean metric is:

$$\bar{f}_{Euc}(x) = \frac{1}{k} \sum_{i=0}^k f^i(x) = \sum_{n=0}^N \left(\frac{1}{k} \sum_{i=0}^k e^{-\lambda_n t_k} \right) c_n \phi_n(x).$$

In general, this image will not be in the orbit of f_0 since one may not be able to find a t^* such that $\frac{1}{k} \sum_{i=1}^k e^{-\lambda_n t_k} = e^{-\lambda_n t^*}$ for all n 's. In contrast, since the average under d_κ is computed for a fixed κ , all the images are mapped to the same section S_κ . Also, since all the images come from the same orbit in this case, they will actually map to the same point on S_κ and their average will be that same point. Thus, our sample mean of images from the same orbit lies in that orbit.

II. Mean of images at the same blur level: Suppose now we have images $f_1(x), f_2(x), \dots, f_k(x)$, with $f_i(x) = \sum_{n=1}^N c_n^i \phi_n(x), i = 1, 2, \dots, k$ that are at the same blur level. That is, $G(f_i(x)) = \sum_{n=0}^N \lambda_n (c_n^i)^2 = \kappa$ for all i . Their Euclidean average is given by:

$$\bar{f}_{Euc}(x) = \frac{1}{k} \sum_{i=1}^k f_i(x) = \sum_{n=0}^N \left(\frac{1}{k} \sum_{i=1}^k c_n^i \right) \phi_n(x).$$

In general $G(\bar{f}_{Euc}(x)) = \sum_{n=0}^N \lambda_n \left(\frac{1}{k} \sum_{i=1}^k c_n^i \right)^2 \neq \kappa$, therefore the Euclidean mean may be at a different blur level than the original images. In our framework, the average under d_κ is calculated on the section S_κ , which ensures that the average image is at the same blur level as the original images. Fig. 18 (a) depicts the difference of our mean versus Euclidean mean for images from the same orbit. And (b) illustrates the difference between Euclidean mean and our mean for images at the same blur level.

APPENDIX

A. Geodesic Paths on Ellipsoidal Sections

Given any two points in a section S_κ , representing two observed signals at the same blurring level, we need to quantify the differences between them. We will use the geodesic distance between them in S_κ , under the Euclidean Riemannian

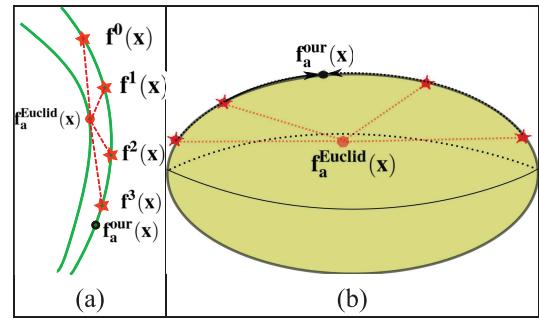


Fig. 18. Comparing Euclidean mean and our mean of images in two different scenarios. (a) Average of images from the same orbit, (b) Average of images at equal blur level.

metric, to define and compute such differences. Let $p_1, p_2 \in S_\kappa$ be two given points and we want to compute a geodesic path between them in S_κ under the chosen metric. Even though S_κ is an ellipsoid, there are no analytical expressions available for geodesics in the high dimensions. We will use a numerical approach, called *path-straightening algorithm* first introduced in [22], but adapted here to the geometry of S_κ . The main idea is to initialize a path α between the two points p_1, p_2 and iteratively straighten it, using the gradient of the energy function $E[\alpha] = \frac{1}{2} \int_0^1 \langle \dot{\alpha}(\tau), \dot{\alpha}(\tau) \rangle d\tau$, until it cannot be straightened any further. The resulting path with minimized energy will be a geodesic. The advantage of this method is that we have a closed form of the gradient of E such that the overall procedure is quite efficient. As described in [22], we first need the followings to construct the geodesic algorithm.

- 1) **Projection onto Mainfold S_κ :** For any arbitrary point $c \in \mathbb{R}^N$, we need a tool to project c to the nearest point in S_κ . One can find this nearest point by iteratively updating c : $c \mapsto c + (\kappa - \tilde{G}(c)) \mathbf{u}_c$, until $\tilde{G}(c) = \kappa$.
- 2) **Projection onto the Tangent Space $T_c(S_\kappa)$:** Given a vector $w \in \mathbb{R}^N$, we need to project w onto $T_c(S_\kappa)$. Since the unit normal to S_κ at c is \mathbf{u}_c , the projection of w on $T_c(S_\kappa)$ is given by $w \rightarrow (w - \langle w, \mathbf{u}_c \rangle \mathbf{u}_c)$.
- 3) **Covariant Derivative and Integral:** Let α be a given path on S_κ , i.e. $\alpha: [0, 1] \rightarrow S_\kappa$, and let w be a vector field along α , i.e. for each $\tau \in [0, 1]$, $w(\tau) \in T_{\alpha(\tau)}(S_\kappa)$. We define the covariant derivative of w along α , denoted $\frac{Dw}{d\tau}$, to be the vector field obtained by projecting $\frac{Dw}{d\tau}(\tau) \in \mathbb{R}^N$ onto the tangent space $T_{\alpha(\tau)}(S_\kappa)$. Covariant integral is the inverse procedure of covariant derivative. A vector field u is called a covariant integral of w along α if the covariant derivative of u is w , i.e. $\frac{Du}{d\tau} = w$. Using the previous item on projection, one can derive tools for computing covariant derivatives and integrals of any given vector field.
- 4) **Parallel Translation:** We also need tools for forward and backward parallel translation of tangent vectors along a given path α on S_κ . A forward parallel translation of a tangent vector $w \in T_{\alpha(0)}(S_\kappa)$, is a vector field along α , denoted \tilde{w} , such that the covariant derivative of \tilde{w} is 0 for all $\tau \in [0, 1]$, i.e. $\frac{D\tilde{w}(\tau)}{d\tau} = 0$, and $\tilde{w}(0) = w$. Similarly, backward parallel translation of

a tangent vector $w \in T_{\alpha(1)}(S_\kappa)$, satisfies that $\tilde{w}(1) = w$ and $\frac{D\tilde{w}(\tau)}{d\tau} = 0$ for all $\tau \in [0, 1]$.

With these tools, we can present the algorithm for computing a geodesic with the path straightening approach. Let's assume a uniform partition $\{0, \frac{1}{k}, \frac{2}{k}, \dots, 1\}$ of $[0, 1]$ for the implementation purpose.

Algorithm (Path Straightening in S_κ): Given two points p_1 and p_2 in S_κ . Suppose $p_1, p_2 \in \mathbb{R}^N$, and $\tau = 0, 1, 2, \dots, k$.

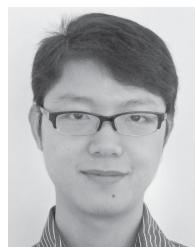
- 1) Initialize a path α : for all $\tau = 0, 1, 2, \dots, k$, using a straight line $(\tau/k)p_1 + (1 - (\tau/k))p_2$ in \mathbb{R}^N . Project each of these points to their nearest points in S_κ to obtain $\alpha(\tau/k)$ using item 1.
- 2) Compute $\frac{da}{d\tau}$ along α : let $\tau = 1, 2, \dots, k$ and $v(0) = \mathbf{0}$. Compute $v(\tau/k) = k(\alpha(\tau/k) - \alpha((\tau-1)/k))$ in \mathbb{R}^N . Project $v(\tau/k)$ into $T_{\alpha(\tau/k)}(S_\kappa)$ using item 2 to get $\frac{da}{d\tau}(\tau/k)$.
- 3) Compute covariant integral of $\frac{da}{d\tau}$, with zero initial condition, along α to obtain a vector field u along α .
- 4) Backward parallel translate $u(1)$ along α to obtain \tilde{u} .
- 5) Compute gradient vector field of E according to $w(\tau/k) = u(\tau/k) - (\tau/k)(\tilde{u}(\tau/k))$ for all τ .
- 6) Update path $\tilde{\alpha}(\tau/k) = \alpha(\tau/k) - \epsilon w(\tau/k)$ by selecting a small $\epsilon > 0$. Then project $\tilde{\alpha}(\tau/k)$ to S_κ to obtain the updated path $\alpha(\tau/k)$.
- 7) Return to step 2 unless $\|w\|$ is small enough or max iteration times reached.

ACKNOWLEDGMENT

The authors would like to thank Prof. Jan Flusser of UTIA for sharing astronomy images and creators of other databases used in this paper for making them available to them.

REFERENCES

- [1] B. Basile, A. Blake, and A. Zisserman, "Motion deblurring and super-resolution from an image sequence," in *Proc. ECCV*, 1996, pp. 573–582.
- [2] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *IEEE Signal Process. Mag.*, vol. 13, no. 3, pp. 43–64, May 1996.
- [3] Y. Lu, J. Sun, L. Quan, and H.-Y. Shum, "Image deblurring with blurred/noisy image pairs," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 1–10, 2007.
- [4] N. Joshi, C. L. Zitnick, R. Szeliski, and D. J. Kriegman, "Image deblurring and denoising using color priors," in *Proc. IEEE Conf. Comput. Pattern Recognit.*, Jun. 2009, pp. 1550–1557.
- [5] J. Portilla, M. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.
- [6] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.
- [7] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *Proc. IEEE Conf. Comput. Pattern Recognit.*, Jun. 2009, pp. 1964–1971.
- [8] J. Ni, P. Turaga, V. M. Patel, and R. Chellappa, "Example-driven manifold priors for image deconvolution," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3086–3096, Nov. 2011.
- [9] H. Hu and G. de Haan, "Low cost robust blur estimator," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2011, pp. 617–620.
- [10] F. Crete, T. Dolmire, P. Ladret, and M. Nicolas, "The blur effect: Perception and estimation with a new no-reference perceptual blur metric," *Proc. SPIE*, vol. 6492, Jan. 2007.
- [11] J. Flusser and T. Suk, "Degraded image analysis: An invariant approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 6, pp. 590–603, Jun. 1998.
- [12] T. Suk and J. Flusser, "Combined blur and affine moment invariants and their use in pattern recognition," *Pattern Recognit.*, vol. 36, no. 12, pp. 2895–2907, 2003.
- [13] J. Kautsky and J. Flusser, "Blur invariants constructed from arbitrary moments," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3606–3611, Dec. 2011.
- [14] B. Chen, H. Shu, H. Zhang, G. Coatrieux, L. Luo, and J. L. Coatrieux, "Combined invariants to similarity transformation and to blur using orthogonal Zernike moments," *IEEE Trans. Image Process.*, vol. 20, no. 2, pp. 345–360, Feb. 2011.
- [15] C.-Y. Wee and R. Paramesran, "Derivation of blur-invariant features using orthogonal Legendre moments," *IET Comput. Vis.*, vol. 1, no. 2, pp. 66–77, Jun. 2007.
- [16] R. Gopalan, S. Taheri, P. Turaga, and R. Chellappa, "A blur-robust descriptor with applications to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1220–1226, Jun. 2012.
- [17] E. Kee, S. Paris, S. Chen, and J. Wang, "Modeling and removing spatially-varying optical blur," in *Proc. IEEE Int. Conf. Comput. Photography*, Apr. 2011, pp. 1–8.
- [18] Z. Zhang, E. Klassen, A. Srivastava, P. K. Turaga, and R. Chellappa, "Blurring-invariant riemannian metrics for comparing signals and images," in *Proc. ICCV*, 2011, pp. 1770–1775.
- [19] I. Makaremi and M. Ahmadi, "Wavelet-domain blur invariants for image analysis," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 996–1006, Mar. 2012.
- [20] R. Hummela, B. Kimia, and S. Zucker, "Gaussian blur and the heat equation: Forward and inverse solutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jan. 1985, pp. 668–671.
- [21] W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*. New York, NY, USA: Academic Press, 1975.
- [22] E. Klassen and A. Srivastava, "Geodesics between 3D closed curves using path-straightening," in *Proc. ECCV, Lect. Notes Comput. Sci.*, 2006, pp. 95–106.
- [23] H. Sheikh, M. Sabir, and A. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3440–3451, Nov. 2006.
- [24] E. C. Larson and D. M. Chandler, "Most apparent distortion: Full-reference image quality assessment and the role of strategy," *J. Electron. Imag.*, vol. 19, no. 1, p. 011006, 2010.
- [25] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2003, pp. 409–415.
- [26] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, "The amsterdam library of object images," *Int. J. Comput. Vis.*, vol. 61, no. 1, pp. 103–112, 2005.
- [27] U. Amsterdam, (2005). *Amsterdam Library of Object Images* [Online]. Available: <http://staff.science.uva.nl/~aloi>
- [28] S. Al-ameri, N. Kalyankar, and S. Khamitkar, "Deblurred Gaussian blurred images," *J. Comput.*, vol. 2, no. 4, pp. 33–35, 2010.
- [29] K. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.



Zhengwu Zhang received the B.S. degree in electrical engineering from the South China University of Technology, Guangzhou, China, in 2008, where he was a participant of Talented Student Program. He received the M.S. degree in school information science & technology from Sun Yat-Sen University, Guangzhou, in 2010. He is currently pursuing the Ph.D. degree with Florida State University, Tallahassee, FL, USA.



Eric Klassen received the Ph.D. degree in mathematics from Cornell University, Ithaca, NY, USA, in 1987. He is a Professor of mathematics with Florida State University, Tallahassee, FL, USA. His current research interests include 3D topology, gauge theory, Riemann surfaces, and computer image analysis.



Anuj Srivastava received the M.S. and Ph.D. degrees in electrical engineering from Washington University, St. Louis, MI, USA, in 1993 and 1996, respectively. From 1996 to 1997 with Brown University, Providence, RI, USA, as a Visiting Researcher, he joined the Department of Statistics, Florida State University, Tallahassee, FL, USA, as an Assistant Professor in 1997. He is currently a Professor with the same department. He has held visiting appointments with the University of Lille, Lille, France, INRIA, Sophia Antipolis, France, and UCL,

Louvain-Le-Neuve, Belgium. His current research interests include pattern theoretic approaches to problems in image analysis, computer vision, and signal processing. He has developed computational tools for performing statistical inferences on certain nonlinear manifolds, including shape manifolds of curves and surfaces. He has published more than 170 peer-reviewed journal and conference articles.