Example-Driven Manifold Priors For Image Deconvolution

Jie Ni, *Student Member, IEEE,* Pavan Turaga, *Member, IEEE,* Vishal M. Patel, *Member, IEEE,* and Rama Chellappa, *Fellow, IEEE*

***Abstract*—Image restoration methods that exploit prior information about images to be estimated have been extensively studied, typically using the Bayesian framework. In this paper, we consider the role of prior knowledge of the object class in the form of a patch manifold to address the deconvolution problem. Specifically, we incorporate unlabeled image data of the object class, say natural images, in the form of a patch-manifold prior for the object class. The manifold prior is implicitly estimated from the given unlabeled data. We show how the patch manifold prior effectively exploits the available sample class data for regularizing the deblurring problem. Furthermore, we derive a generalized cross validation (GCV) function to automatically determine the regularization parameter at each iteration without explicitly knowing the noise variance. Extensive experiments show that this method performs better than many competitive image deconvolution methods.**

***Index Terms*—Deconvolution, patch manifold, local manifold, regularization, generalized cross validation.**

I. Introduction

Image deconvolution is a classical inverse problem where we observe a two-dimensional image y that consists of an unknown desired image x degraded by a point spread function (PSF) h (often assumed to be known) and then corrupted by zero-mean additive white Gaussian noise (AWGN) γ with variance σ^2 (see Fig. 1). Assuming that the images are of size $M \times M$, this model can be expressed as

$$y(n_1, n_2) = (x \circledast h)(n_1, n_2) + \gamma(n_1, n_2), \qquad (1)$$

where $0 \le n_1, n_2 \le M - 1$. Using matrix notation, this model can be written as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \gamma, \qquad (2)$$

where $\mathbf{y}, \mathbf{x}$, and γ are $M^2 \times 1$ lexicographically ordered column vectors representing the arrays y, x and γ, respectively and $\mathbf{H}$ is the $M^2 \times M^2$ matrix that models the point spread function. In the discrete Fourier transform (DFT) domain, we have for (1)

$$Y(k_1, k_2) = H(k_1, k_2)X(k_1, k_2) + \Gamma(k_1, k_2), \qquad (3)$$

where $Y(k_1, k_2), H(k_1, k_2), X(k_1, k_2)$ and $\Gamma(k_1, k_2)$ are the 2D DFTs of y, h, x, and γ, respectively, for $-M/2 \le k_1, k_2 \le M/2 - 1$. Given y and h, we seek to estimate x. Such linear inverse problems often arise in many image processing applications such as radiometry, satellite imaging, optical systems, magnetic resonance imaging and seismic processing.

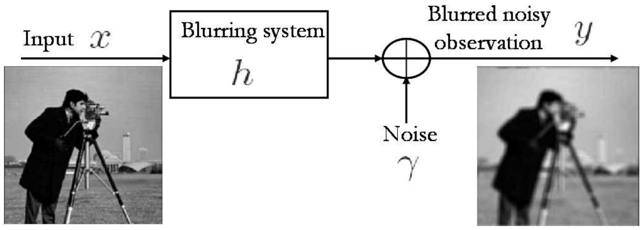


Fig. 1. Model for the deconvolution problem.

It is well known that the deconvolution problem is ill-posed. To find a unique and stable solution, regularization is often used. A popular way to estimate the unknown image x is to use Tikhonov regularization [2] which consists of minimizing the following term

$$J_T(x) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda E(\mathbf{x}), \qquad (4)$$

where $E(\mathbf{x}) = \|\mathbf{C}\mathbf{x}\|_2^2$ and $\mathbf{C}$ is an $M^2 \times M^2$ matrix operator, known as the regularizing operator (e.g. Laplacian). The first term in (4) expresses the fidelity to the data x, and the second term expresses the desired smoothness of the restored image. Here, λ is the regularization parameter that represents the trade-off between fidelity to the data and the smoothness of the recovered image. The solution to Tikhonov regularization problem can be obtained directly in Fourier space

$$\tilde{X}(k_1, k_2) = \frac{H^*(k_1, k_2)Y(k_1, k_2)}{|H(k_1, k_2)|^2 + \lambda |C(k_1, k_2)|^2}. \qquad (5)$$

The Tikhonov method offers computational advantages. However, it often creates Gibbs oscillations in the neighborhood of discontinuities in the image [3]. As a result, the visual quality of the recovered image often degrades.

Recently, considerable effort has been spent on designing alternative sparsity constraints which preserve such features. Methods based on these sparsity constraints have been successfully used for image deconvolution (c.f. [4], [5], [6], [7], [8], [9], [10], [11]). Transformations such as wavelets, curvelets [12], contourlets [13], [14] and shearlets [15] are popular for image representation and are often used for image restoration.

J. Ni, P. Turaga, V. M. Patel and R. Chellappa are with the Department of Electrical and Computer Engineering and the Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742 (e-mail: {jni,turaga,pvishalm,rama}@umiacs.umd.edu)

A preliminary version of this paper appeared in DIPA 2010 [1]. This work was supported by an ONR MURI Grant N00014-08-1-0638.

EDICS: TEC-RST, SMR-SMD

However, it has been shown that for image restoration, learning representation from examples instead of using pre-specified ones, usually leads to improved results [16], [17], [18], [19].

Another popular deconvolution method is based on total variation [20], where $E(\mathbf{x})$ in (4) is set equal to $\|\mathbf{Cx}\|_1$, where $\|\mathbf{Cx}\|_1$ is the ℓ_1-norm of gradients of $\mathbf{x}$. Variations of this method have also been proposed [21], [22]. A local polynomial approximation method that uses intersecting confidence intervals was proposed in [23]. In [24], a locally adaptive kernel regression method was proposed to solve (4).

We take a different approach to the problem of image deconvolution by exploiting extra information in the form of prior knowledge of the object class to regularize the inverse problem [25]. Specifically, we use image data of the object class, as the available extra information. The proposed method assumes that the set of all patches (e.g. 3×3) from a given class of images - say faces, or natural images - live on a *manifold*. We shall define this in more precise terms as we progress. First, let us motivate the role of the patch-manifold in representing images. Images are formed by the interaction of light with surfaces. Surface properties such as geometry and reflectance give rise to varied appearances, which are then imaged by a projective camera. To characterize the space of images thus formed, one needs to have a clear model for each of these factors. For example, under variations in lighting conditions, with fixed viewing angle and pose, the set of face-images obtained lives on a ‘cone’ [26]. However, it is difficult to extend these results to more general classes of objects and scenes. Alternately, vision researchers have explored the tools of ‘manifold-learning’ in such cases when one may have access to a large set of examples from each class. Manifold learning algorithms such as Isomaps [27], LLE [28] etc, have proven useful in many cases and have been used to estimate the manifold of faces under pose variations. However, image manifolds are extremely high dimensional in the general case, since real images result from all of the above factors playing out simultaneously instead of in isolation. The situation gets much more complicated when several objects are present in the scene, each with its own surface properties. Since the number of samples needed to estimate even relatively low-dimensional manifolds is quite high (c.f. [29]), this makes estimating image-manifolds in a general unconstrained setting, a difficult proposition.

On the other hand, assuming that small patches from a given class lie on a manifold is a far weaker requirement. It can be shown that even simple patch-manifold models give rise to complex imagery. For example, by assuming that each patch consists of small binary line-segments, one can span the set of all ‘cartoon’-images. Similarly, the patch-manifold of locally parallel textures gives rise to complex finger-print type images [25]. Locally parallel textures can be analytically described by 2D sinusoidal functions, whereas the global manifold of images thus obtained is hard to describe in closed-form. For the case, when one does not know an analytical form for the patch manifold, patch-manifold learning is still far easier than image-manifold learning. Since even a single image gives rise to an abundance of patches, this affords the availability of an extremely large set of samples on the patch manifold from unlabeled data. Coupled with the fact that the space of patches is far smaller than the space of images, this makes estimating the patch-manifold far easier.

This paper makes the following contributions: 1) We show how the patch manifold prior effectively exploits the available sample data for regularizing the deblurring problem. 2) We present efficient parameterizations for learning and using the patch-manifold. 3) We derive a closed-form generalized cross validation function to automatically find a value of the regularizer λ without explicitly calculating the noise variance.

A. Organization of the Paper

The rest of the paper is organized as follows. In section II, we discuss the fundamentals of the patch-manifold and how it is used for regularizing the deblurring problem. In section III, we present parameterizations and efficient algorithms for computing the parameters of the patch manifold. Then, we derive a generalized cross-validation function for the proposed method in section IV. In section V, we present experimental results. Finally, discussions and concluding remarks are given in section VI.

II. Manifold Modeling of Image Classes

In the following, we use x to denote the unknown image to be solved for, and $\mathbf{x}$ as the vector representation of the image x. We follow the theoretical foundations set forth in [25] for modeling images using a patch-manifold. We briefly review the required preliminaries before describing how we employ it for the deblurring problem. Let us denote a patch extracted from the image x, at location $q \in [0,1]^2$ of width $\tau > 0$ by $p_q(x)(t) = x(q+t) \forall t \in [-\tau/2, \tau/2]^2$. The image $x \in L^2[0,1]^2$, i.e. the set of 2-dimensional finite energy signals. The class dependent image-ensemble is then $\Theta \subset L^2[0,1]^2$. The patch-manifold associated with this ensemble is $\mathcal{M} = \{p_q(x) | q \in [0,1]^2, x \in \Theta\} \subset L^2[-\tau/2, \tau/2]^2$. An image x is now represented as a surface traced on the manifold $\mathcal{M}$ given as

$$c_x : q \mapsto p_q(x) \in \mathcal{M}. \tag{6}$$

Given an image and the manifold representation, one can now measure the goodness of fit between them. To do this, first one needs a way to compute the closest point on the manifold. This is done in two stages. First, patches from an image are projected onto the patch-manifold. This step is denoted by $c(q) = Proj_{\mathcal{M}}(p_q(x))$, which assigns closest patches from the manifold to the given image patches. Thus, $Proj_{\mathcal{M}}(p) = \underset{t \in \mathcal{M}}{\arg\min} \|p - t\|$. The distance of a patch from the manifold is then given by $d(p, \mathcal{M}) = \|p - Proj_{\mathcal{M}}(p)\|$. Then, the goodness of fit of a given image is measured by averaging the distance of each patch from the patch-manifold

$$E_{\mathcal{M}}(x) = \int_{[0,1]^2} d(p_q(x), \mathcal{M})^2 dq \tag{7}$$

$$= \int_{[0,1]^2} \|p_q(x) - Proj_{\mathcal{M}}(p_q(x))\|^2 dq. \tag{8}$$

An image x has low-energy $E_{\mathcal{M}}(x)$ if it traces a curve $c_x = \{p_q(x)\}$ close to the manifold. This curve can be projected onto the manifold by means of the $Proj$ operator. The projected curve is thus represented as

$$\widetilde{c}_x(q) = Proj_{\mathcal{M}}(p_q(x)) \in \mathcal{M}. \tag{9}$$

Now, from this projected curve one can compute the projection of the image x onto the set of images generated by the patch manifold. Reconstruction is achieved by means of averaging overlapping patches. Specifically, the projection of the image x is represented by $Proj_{\mathcal{M}}(x) = Aver(\widetilde{c}_x)$, where,

$$Aver(c_x) = \frac{1}{\tau^2}\int_{\|q-z\|\leq\tau/2} p_z(x-z)dz, \text{with } p_z(c) = c(z). \tag{10}$$

A. Regularizing the deblurring problem with the manifold prior

The optimization problem for deblurring is now recast by introducing a new variable c^* which is a manifold-valued function. The optimization is rewritten as finding an optimal $\mathbf{x}^*$, given an observation y and the manifold prior as

$$\begin{aligned}(x^*, c^*) &= E(x, c) & (11)\\ &= \underset{x,c}{\arg\min} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \lambda \int_{[0,1]^2} \|p_q(x) - c(q)\|^2 dq & (12)\end{aligned}$$

where λ controls the relative weightage between the data and prior terms. A stationary point is obtained by means of an iterative procedure that alternates between solving for $\mathbf{x}^*$ and c^*. Given a current estimate of the image $x^{(k)}$, $c^{(k)}$ is obtained as

$$c^{(k+1)} = Proj_{\mathcal{M}}(x^{(k)}). \tag{13}$$

Next, given $c^{(k+1)}$, we solve for $\mathbf{x}$ as

$$\mathbf{x}^{(k+1)} = (\mathbf{H^T H} + \lambda \mathbf{I})^{-1}(\mathbf{H^T y} + \lambda \mathbf{vec}(Aver(c^{(k+1)})), \tag{14}$$

where $Aver(c)$ is as defined in (10), and $\mathbf{vec}()$ returns the vectorized version of its argument. This procedure is repeated till convergence. This procedure is summarized in Table I.

As (12) is non-convex, the algorithm in Table I may not converge to the global minimizer. However, for a smooth manifold $\mathcal{M}$, the iterates $(x(k), c(k))$ of our algorithm will converge to a stationary point $(x^*$,$c^*)$ [25]. Note that the matrix inversions involved in the optimization steps in Table I are all implemented implicitly using the properties of the PSF matrix $\mathbf{H}$ [30].

TABLE I
ALGORITHM FOR PATCH-MANIFOLD BASED REGULARIZATION FOR DEBLURRING [25].

1. Set $\mathbf{x}^{(0)} = \mathbf{H^T y}$ and $k \leftarrow 0$.
2. Update the manifold-valued function as
$$\forall q \in [0,1]^2, c^{(k+1)}(q) = Proj_{\mathcal{M}}(p_q(\mathbf{x}^{(k)})).$$
3. Update the current estimate of $\mathbf{x}$ as
$$\mathbf{x}^{(k+1)} = (\mathbf{H^T H} + \lambda \mathbf{I})^{-1}(\mathbf{H^T y} + \lambda \mathbf{vec}(Aver(c^{(k+1)}))).$$
4. Repeat till convergence or till maximum iterations are reached.

III. SAMPLING AND LEARNING THE PATCH-MANIFOLD

In actual implementation, we do not have an analytical characterization of the patch manifold. An analytical characterization would lead to a closed-form version of the $Proj$ operator. We instead learn the manifold using training examples of images from the class of images under consideration, e.g. faces or natural images. The $Proj$ operation then amounts to searching for the closest point to a given patch in the learnt manifold. We explore two ways to solve this problem - non-parametric and parametric. We describe these next.

A. Non-parametric manifold learning

In the non-parametric case, we assume that we have a large number of samples from the underlying patch-manifold. In experiments we find that the assumption of a dense sampling is infact very well justified given the easy availability of a large number of patches. With this, the $Proj$ operation is efficiently implemented using approximate nearest neighbor search strategies. We choose locality sensitive hashing (LSH) [31] for this task due to its sub-linear search efficiency. Given a training set of images, patches centered at all pixel locations are extracted from every image. The set of patches thus obtained constitutes the sampling of the manifold. This set is then indexed using LSH.

Here, we briefly review the basic concepts of LSH. LSH attempts to solve a problem called the (r,ϵ)-NN problem. The problem is described as follows: Given a database of points $D = \{x_i\}$ in $\mathbb{R}^n$, and a query x_q, if there exists a point $x \in D$ such that $d(x, x_q) \leq r$, then with high-probability, a point $x' \in D$ is retrieved such that $d(x', x_q) \leq (1+\epsilon)r$. Now, LSH solves this problem by constructing a family of hash-functions $\mathcal{F}$ over $\mathbb{R}^n$ called locality-sensitive, if for any $u, v \in D$

$$\begin{aligned}d(u,v) \leq r &\Rightarrow Pr(f(u) = f(v)) \geq p_1 & (15)\\ d(u,v) \geq (1+\epsilon)r &\Rightarrow Pr(f(u) = f(v)) \leq p_2 & (16)\end{aligned}$$

Popular choices of f include random projections i.e. $f(v) = sgn(v.r)$ where r is a randomly chosen unit-vector, and sgn is the signum function. In this case, f is binary valued taking values in $\{+1, -1\}$. A generalization of this is termed random projections using ‘p-stable’ distributions [32], where $f(v) = \lfloor \frac{v.r+b}{w} \rfloor$ where r is a randomly chosen direction whose entries are chosen independently from a stable distribution, and b is a

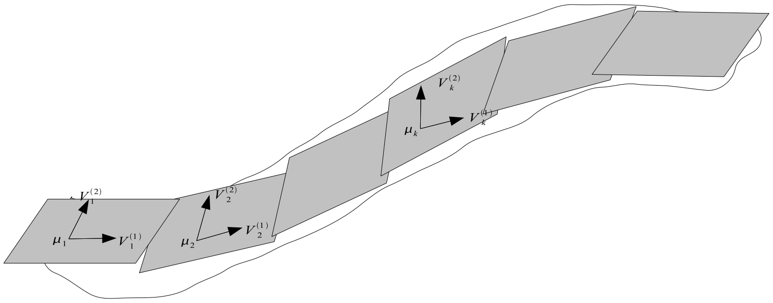


Fig. 2. Locally-linear parametrization of a densely sampled manifold.

random number chosen between $[0, w]$. In this case, the hash function takes on integer values. A $k-$bit hash is constructed by appending k randomly chosen hash-functions

$$F(x) = [f_1(x), f_2(x), \dots f_k(x)] \tag{17}$$

Thus, $F \in \mathcal{F}^k$. Then, L hash tables are constructed by randomly choosing $F_1, F_2 \dots F_L \in \mathcal{F}^k$. All the training examples (patches) are hashed into the L hash tables. For a query point x_q, an exhaustive search is carried out among the examples in the union of the L hash-buckets indexed by q. Appropriate choices of k and L ensure that the algorithm succeeds in finding a (r, ϵ)-NN of the query x_q with a high probability. In our work, we used random projections based hashing i.e. the hash function is $f(v) = sgn(v.r)$.

B. Parametric manifold learning

Even though a dense sampling of the patch-manifold appears to be a reasonable assumption, implementing the $Proj$ operation involves significant computation for the entire image, as we need to hash and search for every patch in the given image. Also, the $Proj$ operator implemented in this manner is susceptible to noise in the dataset. Further, if the sampling density is reduced, the quality of reconstructions can be significantly affected. To deal with these situations, we explore a parametric way for modeling the patch manifold. While several parameterizations of the patch-manifold are possible, we choose the one that leads to computationally efficient algorithms for implementing the $Proj$ operation. Note that one could potentially use algorithms such as LLE [28] and Isomaps [27] to estimate the manifold, but there are a few considerations which make their use prohibitive in the current setting. To begin with, these algorithms have a high computational complexity for estimating the manifold when number of samples is high. Further, out-of-sample extension i.e. finding the parameters of a new patch which is not in the training database, is a non-trivial task [33]. Here, we propose a much simpler parametrization of the patch-manifold which is computationally efficient to learn even when a dense sampling of the patch-manifold is available, and has a graceful out-of-sample extension even when the sampling density reduces.

We assume that the patch manifold can be decomposed into a union of subspaces i.e. $\mathcal{M} = \bigcup_{i=1}^{K} S_i$, where each S_i is a d-dimensional affine subspace in $\mathbb{R}^n$, represented by its offset μ_i and orthonormal basis vectors $\mathbf{V}_i$ (written in matrix form). Each patch on the manifold is then parameterized by the index of the subspace on which it lies and the coefficients of its projection on the appropriate subspace. i.e. the parametrization of a patch p is given by

$$\psi(p) = (\hat{i}, \hat{\alpha}) = \arg\min_{i,\alpha} \|p - \mu_i - \mathbf{V}_i \alpha\| \, . \tag{18}$$

Figure 2 presents a graphical illustration of the locally-linear parametrization of the manifold. To learn this manifold from the training data, we adopt a two stage approach. In the first stage, given the training set of patches $D = \{x_i\}$, we cluster all the patches into K distinct clusters. Each cluster center is associated with the offset of the subspaces μ_i. Within each cluster, we then estimate optimal basis vectors using principal component analysis (PCA). Given a new patch, the closest patch on the manifold is estimated in two stages. In the first stage, the closest cluster center is computed by comparing it with all the cluster centers. Once the closest cluster center is found, the patch is projected onto the subspace of that cluster. i.e. given a new patch p

$$\hat{i} = \min_i \|p - \mu_i\| \, , \quad \hat{\alpha} = \mathbf{V}_{\hat{i}}^T (p - \mu_{\hat{i}}) \tag{19}$$

Then, the $Proj$ operation is easily implemented as

$$Proj_{\mathcal{M}}(p) = \mu_{\hat{i}} + \mathbf{V}_{\hat{i}} \hat{\alpha}, \tag{20}$$

where $(\hat{i}, \hat{\alpha})$ are as defined in (19).

IV. Generalized Cross Validation (GCV)

Note that the deblurring cost function in (12) and thereby the solution in (14) depends on the choice of the regularized λ. The deblurred image depends greatly on the degree of regularization which is determined by the regularization parameter [30]. In this section, we describe a generalized cross validation (GCV) function [34], [35] to compute the regularization parameter automatically. The GCV method is based on statistical considerations, namely, that a good value of the regularization parameter should predict missing data values [36]. One of the main advantages of this GCV method is that it obtains the regularization parameter without knowing the noise variance. It depends only on the data and automatically adjusts the regularization parameter according to the data.

First, we define the singular value decomposition (SVD) of the blur matrix $\mathbf{H}$ as $\mathbf{H} = \mathbf{U \Sigma V}^T$, where $\mathbf{U}$ and $\mathbf{V}^T$ are orthogonal matrices, satisfying $\mathbf{U}^T\mathbf{U} = \mathbf{I}_{M^2}$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I}_{M^2}$, and $\mathbf{\Sigma} = diag(\sigma_i)$ is a diagonal matrix. Let $\mathbf{u}_i$ and $\mathbf{v}_i$ be the columns of $\mathbf{U}$ and $\mathbf{V}$, respectively.

In the principle of minimizing the predictive mean-square error, [36] defines the GCV function as

$$G(\lambda) = \frac{\|\mathbf{Hx} - \mathbf{y}\|_2^2}{(trace(\mathbf{I} - \mathbf{HH}^\sharp))^2} \tag{21}$$

where $\mathbf{x}$ is the restored image and $\mathbf{H}^\sharp$ is the regularized inverse given by

$$\begin{aligned} \mathbf{H}^\sharp &= (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^T \\ &= (\mathbf{V\Sigma}^2\mathbf{V}^T + \lambda\mathbf{I})^{-1}\mathbf{V\Sigma U}^T \\ &= \mathbf{V}(\mathbf{\Sigma}^2 + \lambda\mathbf{I})^{-1}\mathbf{\Sigma U}^T. \end{aligned} \tag{22}$$

Let $\phi_i = \frac{\sigma_i^2}{\sigma_i^2+\lambda}$, $\mathbf{\Phi} = diag(\phi_i)$, then (22) can be written as

$$\mathbf{H}^\sharp = \mathbf{V\Phi\Sigma}^{-1}\mathbf{U}^T. \tag{23}$$

Substituting (23) into (21), the GCV function becomes

$$G(\lambda) = \frac{\|\mathbf{y} - \mathbf{Hx}\|_2^2}{(trace(\mathbf{I} - \mathbf{HV\Phi\Sigma}^{-1}\mathbf{U}^T))^2}. \tag{24}$$

By replacing $\mathbf{x}$ in (24) with the manifold-based solution, we obtain the GCV function of our proposed algorithm.

We split the manifold solution into two parts: $\mathbf{x} = \mathbf{x}_\lambda + \tilde{\mathbf{x}}$, where $\mathbf{x}_\lambda = (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^T\mathbf{y} = \mathbf{H}^\sharp\mathbf{y} = \mathbf{V\Phi\Sigma}^{-1}\mathbf{U}^T\mathbf{y}$, $\tilde{\mathbf{x}} = (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I})^{-1}\lambda\mathbf{vec}(Aver(c^{(k)}))$.
Hence, $\mathbf{y} - \mathbf{Hx} = (\mathbf{y} - \mathbf{Hx}_\lambda) - \mathbf{H}\tilde{\mathbf{x}}$, where

$$\begin{aligned} \mathbf{y} - \mathbf{Hx}_\lambda &= \mathbf{y} - \mathbf{HV\Phi\Sigma}^{-1}\mathbf{U}^T\mathbf{y} \\ &= \mathbf{y} - \mathbf{U\Sigma V}^T\mathbf{V\Phi\Sigma}^{-1}\mathbf{U}^T\mathbf{y} \\ &= \mathbf{y} - \mathbf{U\Phi U}^T\mathbf{y}. \end{aligned} \tag{25}$$

Also,

$$\begin{aligned} \mathbf{H}\tilde{\mathbf{x}} &= \mathbf{H}(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I})^{-1}\lambda\mathbf{vec}(Aver(c^{(k)})) \\ &= \mathbf{U\Sigma V}^T\mathbf{V}(\mathbf{\Sigma}^2 + \lambda\mathbf{I})^{-1}\mathbf{V}^T\lambda\mathbf{vec}(Aver(c^{(k)})) \\ &= \mathbf{U\Sigma}(\mathbf{\Sigma}^2 + \lambda\mathbf{I})^{-1}\mathbf{V}^T\lambda\mathbf{vec}(Aver(c^{(k)})) \end{aligned} \tag{26}$$

Since the 2-norm is invariant under orthogonal transformation, $\|\mathbf{y} - \mathbf{Hx}\|_2^2 = \|\mathbf{U}^T(\mathbf{y} - \mathbf{Hx})\|_2^2$, so we can work in the coordinates of the SVD. From (25) and (26), we have

$$\begin{aligned} &\|\mathbf{y} - \mathbf{Hx}\|_2^2 \\ &= \|\mathbf{U}^T(\mathbf{y} - \mathbf{Hx}_\lambda - \mathbf{H}\tilde{\mathbf{x}})\|_2^2 \\ &= \|\mathbf{U}^T(\mathbf{y} - \mathbf{U\Phi U}^T\mathbf{y} - \\ &\mathbf{U\Sigma}(\mathbf{\Sigma}^2 + \lambda\mathbf{I})^{-1}\mathbf{V}^T\lambda\mathbf{vec}(Aver(\mathbf{c}^{(k)})))\|_2^2 \\ &= \|(\mathbf{I} - \mathbf{\Phi})\mathbf{U}^T\mathbf{y} - \mathbf{\Sigma}(\mathbf{\Sigma}^2 + \lambda\mathbf{I})^{-1}\mathbf{V}^T\lambda\mathbf{vec}(Aver(\mathbf{c}^{(k)}))\|_2^2 \\ &= \sum_{i=1}^{M^2}\left(\frac{\lambda\mathbf{u}_i^T\mathbf{y} - \lambda\sigma_i\mathbf{v}_i^T\mathbf{vec}(Aver(c^{(k)}))}{\sigma_i^2 + \lambda}\right)^2. \end{aligned} \tag{27}$$

Further,

$$\begin{aligned} &(trace(\mathbf{I} - \mathbf{HV\Phi\Sigma}^{-1}\mathbf{U}^T))^2 \\ &= (trace(\mathbf{I} - \mathbf{U\Sigma V}^T\mathbf{V\Phi\Sigma}^{-1}\mathbf{U}^T))^2 \\ &= (trace(\mathbf{U}(\mathbf{I} - \mathbf{\Phi})\mathbf{U}^T))^2 \\ &= (trace(\mathbf{I} - \mathbf{\Phi}))^2 \\ &= \left(\sum_{i=1}^{M^2}\frac{\lambda}{\lambda + \sigma_i^2}\right)^2. \end{aligned} \tag{28}$$

Hence, substituting the expressions from (27) and (28) into (24), we obtain the GCV function for our manifold-based algorithm

$$G^{(k)}(\lambda) = \frac{\sum_{i=1}^{M^2}\left(\frac{\mathbf{u}_i^T\mathbf{y} - \sigma_i\mathbf{v}_i^T\mathbf{vec}(Aver(c^{(k)}))}{\sigma_i^2+\lambda}\right)^2}{\left(\sum_{i=1}^{M^2}\frac{1}{\sigma_i^2+\lambda}\right)^2}. \tag{29}$$

Note that the GCV function changes with every iteration and is thus indexed with k. This means that the optimal value of λ changes with every iteration. Hence, at each iteration we need to compute the best λ by evaluating the GCV function for various values of λ and choosing one that minimizes the GCV function. Thus,

$$\lambda_{optimal}^{(k)} = \arg\min_{\lambda} G^{(k)}(\lambda), \tag{30}$$

where $G^{(k)}(\lambda)$ is as given in (29).

V. Experimental Results

In this section, we present the results of our algorithm and compare them with various state-of-the-art methods: deconvolution based on sparsity prior in wavelet domain [9], hyper-Laplacian prior-based deconvolution [37], Fourier-Wavelet Regularized deconvolution (ForWaRD) [6], Anisotropic non-parametric image resotoration (LPA-ICI) [23] and Tikhonov deconvolution [30]. The regularization parameters for these methods are either chosen from a set of values within a wide range or set to be the optimal value reported in the corresponding papers. In the following experiments, we use the improvement in signal-to-noise-ratio (ISNR) to compare the different methods. The ISNR is defined as

$$ISNR = 10\log_{10}\left(\frac{\|\mathbf{x} - \mathbf{y}\|_2^2}{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2}\right).$$

For an image of size $M \times N$, the BSNR is defined in decibels as

$$BSNR = 10\log_{10}\left(\frac{\|\mathbf{Hx} - \mu(\mathbf{Hx})\|_2^2}{MN\sigma^2}\right),$$

where $\mu(\mathbf{Hx})$ represents the mean of $\mathbf{Hx}$.

Fig. 3 shows some of the images used to learn the patch manifold for our algorithm. We randomly sample $22,500$ patches of size 4×4 from each image, so we have $112,500$ patches in total to learn the patch manifold. In Fig. 4, we display the test images used for different experiments in this paper.

In the first set of experiments, a *Barbara* image, shown in Fig. 4(a), is blurred by the following point spread function:

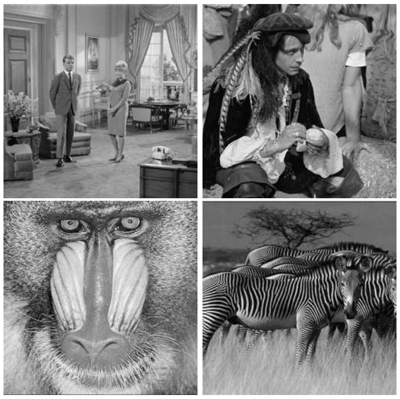
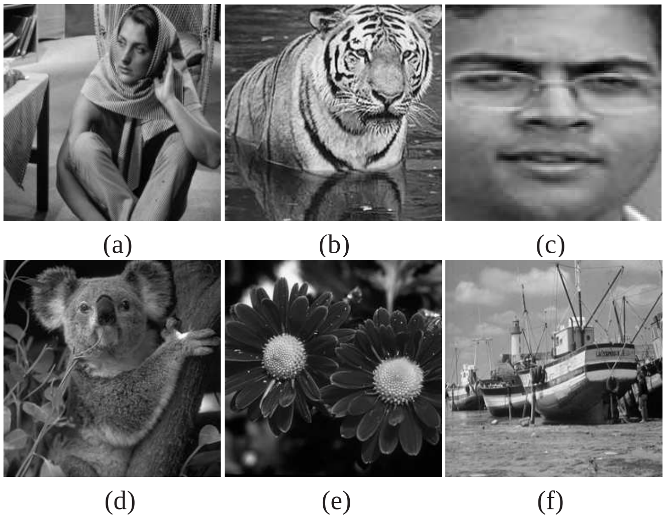


Fig. 3. Some of the natural images used to learn the patch-manifold of natural images.



(a) (b) (c)

(d) (e) (f)

Fig. 4. Images used in this paper for different experiments. (a) *Barbara* image, (b) *Tiger* image, (c) a *face* image, (d) *Koala* image, (e) *Flowers* image and (f) *Boat* image.

$h(n_1, n_2) = (1 + n_1^2 + n_2^2)^{-1}$, for $n_1, n_2 = -7, ..., 7$. The AWGN variance, σ^2, is chosen with a BSNR of 40 dB. The ISNR values obtained by different methods are compared in Table II under the Experiment 1 column. The parametric and non-parametric manifold-based methods yield ISNR values of 7.98 dB and 7.95 dB, respectively which are better than the values obtained by any of the other methods. A portion of the image is zoomed in to reveal the visual detail of the results obtained by the different methods, and are shown in Fig. 5(a)-(f). As can be seen from the figure, our manifold-based method recovers details better than the other methods.

In the second set of experiments, the *Tiger* image, shown in Fig. 4(b), is blurred by a real-world camera shake kernel [38]. In this experiment, we choose the noise variance, σ^2, with a BSNR of 30 dB. The simulation results are reported under the Experiment 2 column of Table II. The deblurred image details obtained by the different methods are shown in Fig. 6(a)-(f). The blur PSF used in this experiment is shown in Fig. 6(g). The LPA-ICI method gives the ISNR value of 9.14 dB which is slightly better than our method. Note that the LPA-ICI method obtains the initial estimate using a local polynomial approximation method. To further enhance their performance, a regularized Wiener filtering (RW) is applied to the initial estimate. Similarly, we can enhance the performance of our algorithm by adapting RW filtering as a postprocessing step as was done in [6] and [8].

In the third set of tests, a *face* image is blurred by a Gaussian PSF defined as

$$h(i, j) = De^{-\frac{i^2+j^2}{2\eta^2}}$$

for $i, j = -5, ..., 5$, where D is a normalizing constant ensuring that the blur is of unit mass, and η^2 is the variance that determines the severity of the blur. Noise is added with a BSNR of 40 dB. The results are summarized under the Experiment 3 column of Table II. Again, our manifold-based algorithm performs the best in terms of ISNR. A portion of the deblurred images from different methods are shown in Fig. 7(a)-(f).

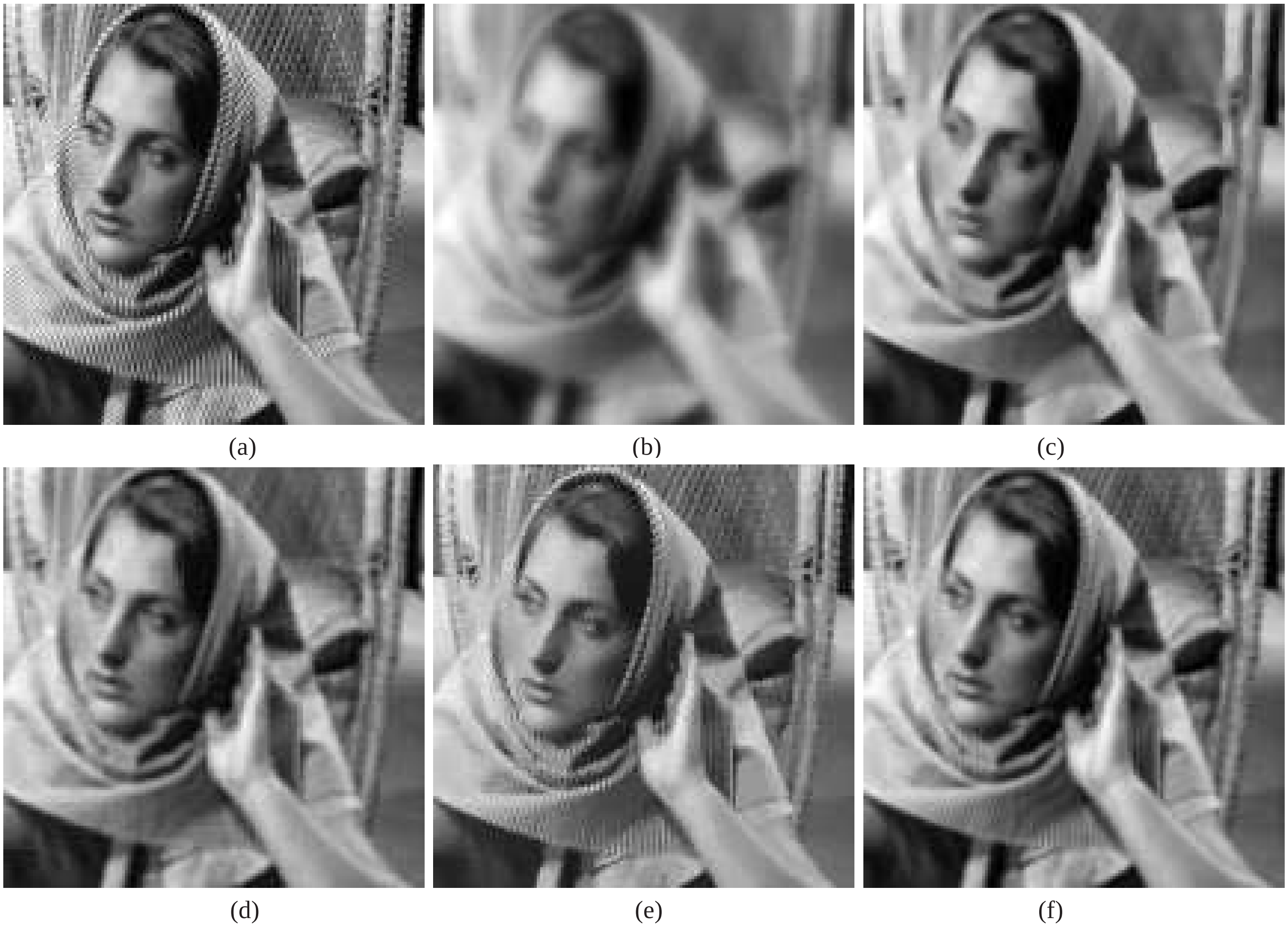
In the fourth set of tests, the image of *Koala* is blurred by a separable filter [23] with weights $[1, 4, 6, 4, 1]/16$ in both the horizontal and vertical directions and the AWGN is added such that the BSNR equal to 30 dB. The simulation results are reported under the Experiment 4 column of Table II. Both wavelet domain sparsity-based method and parametric manifold-based method perform approximately the same with ISNR values of 3.25 dB and 3.21 dB respectively. In this experiment, the non-parametric manifold-based algorithm performed the best with an ISNR value of 3.48 dB.

In the fifth experiment, we apply a horizontal motion blur kernel with length 7 on a *Flowers* image. For this experiment, the BNSR value is set to be 25 dB. The deconvolution results obtained by different methods are reported under the Experiment 5 column of Table II. Both parametric and non-parametric manifold-based methods perform the same yielding ISNR values of 7.65 dB. This experiment shows that, even in the case of low BSNR, our manifold-based method can provide better reconstruction than some of the competitive deconvolution methods.

In Fig. 8(a)-(c), we display a few of the GCV curves obtained from Experiment 1, 4 and 5, respectively. The minimizers of these GCV curves are chosen to be the regularization parameters in each experiment. Hence, unlike some of the other deconvolution algorithms such as [6], our method does not require the explicit knowledge of noise variance and it automatically determines the regularization parameter at each iteration.

In Fig. 9, we compare the ISNR performance of different methods as a function of BSNR. For this experiment, we used the Gaussian blur on the *Barbara* image. As can be seen from the figure, manifold-based method decays slower as a function of noise level given in terms of BSNR.

The stopping criterion for our method is usually that the norm of the difference between two successive estimates falls below a pre-specified threshold. That is, we stop when $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)}\|_2^2 < 10^{-3}$. Empirical results show that our manifold-based methods typically converge in about 3 to 5 iterations. In Fig. 10, we plot $\frac{1}{MN}\|\mathbf{y} - \mathbf{H}\mathbf{x}^{(k)}\|_2^2$ vs. number of iterations curves for the case when a Gaussian blur is applied on the image shown in Fig. 4(c) with BSNR of 35 dB. As can be seen



(a) (b) (c)

(d) (e) (f)

Fig. 5. Details of the image deconvolution experiment with a *Barbara* image. (a) Original image. (b) Noisy blurred image. (c) Hyper-Laplacian [37] estimate (ISNR 5.19 dB). (d) Wavelet domain sparsity-based estimate [9] (ISNR 6.24 dB).(e)LPA-ICI [23] estimate (ISNR 7.88 dB) (f) Parametric manifold-based estimate (ISNR 7.98 dB) suggested in this paper.

TABLE II
ISNR FOR DIFFERENT EXPERIMENTS. THE HIGHEST ISNR FOR EACH EXPERIMENT IS SHOWN IN BOLD.

	Experiments				
Method	1. Barbara	2. Tiger	3. Face	4. Koala	5. Flowers
Non-parametric Manifold-based deconvolution	7.95	8.96	8.27	**3.48**	**7.65**
Parametric Manifold-based deconvolution	**7.98**	9.02	**8.49**	3.21	**7.65**
Anisotropic Nonparametric Image Restoration(LPA-ICI)	7.88	**9.14**	7.40	3.38	5.13
Fourier-Wavelet Regularized Deconvolution(ForWaRD)	7.6	9.02	7.74	3.04	7.4
Wavelet domain sparsity-based deconvolution	6.24	8.28	6.1	3.25	6.03
Hyper-laplacian prior-based deconvolution	5.19	8.14	5.16	2.74	5.39
Tikhonov deconvolution	3.04	4.26	4.39	1.02	4.64

from the figure, our method converges in about 3 iterations and the difference between the two estimates after 3 iterations is very small.

A. Blind Deconvolution

In real applications, we do not have the form of the blur kernel. Hence, this requires blind deconvolution methods. It is stated in [38] that a robust blind deconvolution strategy is to first use the maximum a-posterior (MAP) estimate to recover the blur kernel, and then use the recovered kernel to solve for the sharp image using non-blind deconvolution algorithm. In this experiment, we employ this strategy and test the robustness of our method to small errors in blur-kernel estimation. We apply a 5×5 box-car blur on an image, shown in Fig. 4(f), with BSNR of 35 dB. Fig. 11 shows the details of blind deconvolution result using the method proposed in [39] and the deconvolution result using parametric manifold method based on the blur kernel estimated by [39]. The ISNR values are -0.19 dB and 1.59 dB, respectively. We observe that our method can suppress the ringing artifacts and is robust when the estimated kernel is not accurate enough.

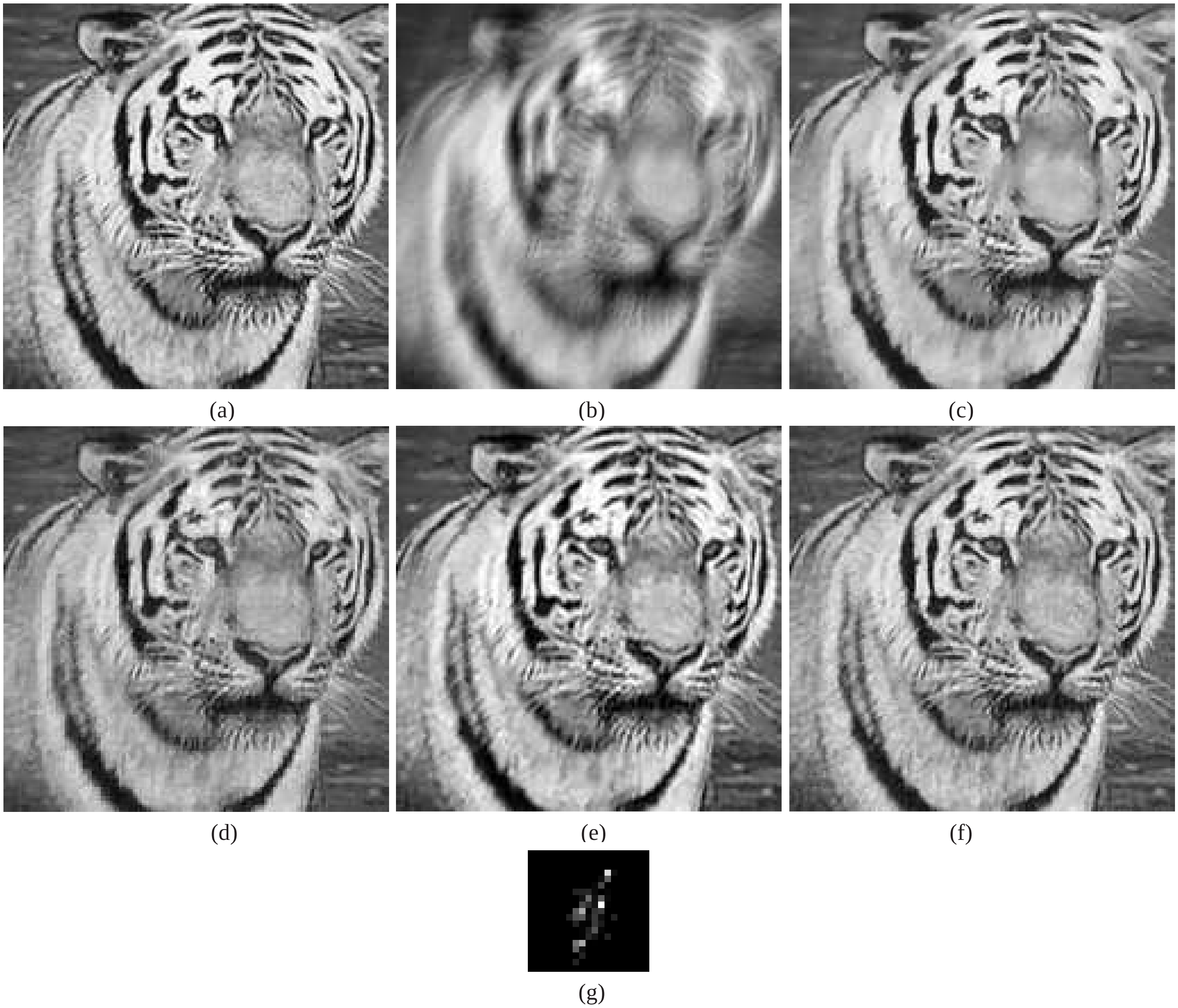


Fig. 6. Details of the image deconvolution experiment with a *Tiger* image. (a) Original image. (b) Noisy blurred image. (c)Hyper-laplacian [37] estimate (ISNR 8.14 dB). (d) Wavelet domain sparsity-based estimate [9] estimate (ISNR 8.28 dB). (e) LPA-ICI [23] estimate (ISNR 9.14 dB) (f) Parametric manifold-based estimate suggested in this paper (ISNR 9.02 dB). (g) Blur kernel.

B. Computational Complexity

In our deconvolution method, the most computationally intensive part is to find the projection onto the manifold. Using Matlab on a linux system with Intel Core 2.00 GHz/2.00 GB processor, projecting one patch onto a manifold formed by 112,500 patches using non-parametric manifold learning takes around 2.5e-2 seconds. Using parametric manifold learning algorithm, projecting one patch onto the patch manifold of the same size takes 5e-3 seconds. On average our algorithm takes about 3 minutes to process an image of size 256×256.

Based on the experimental results, we observe that using the parametric manifold gives similar performance as the non-parametric case, while the former is much more computationally efficient. Further, the computation can be made more efficient by making the sampling of the patch manifold more compact.

VI. Conclusion

In this paper, we have presented a way of utilizing unlabeled image data to regularize the deconvolution problem. We formalized this via a patch-manifold prior for image classes which was shown to work very well for a wide variety of image content. This paves the way for interesting new directions of work. For example, using image formation models for specific cases, one could ask if there exist closed form expressions for the patch manifold. Further, several other inverse problems such as super-resolution, recovery of compressed signals, etc can be explored using example-driven priors. Finally, it would be interesting to fuse the example data with multi-view geometric constraints to better estimate the patch manifold with fewer examples.

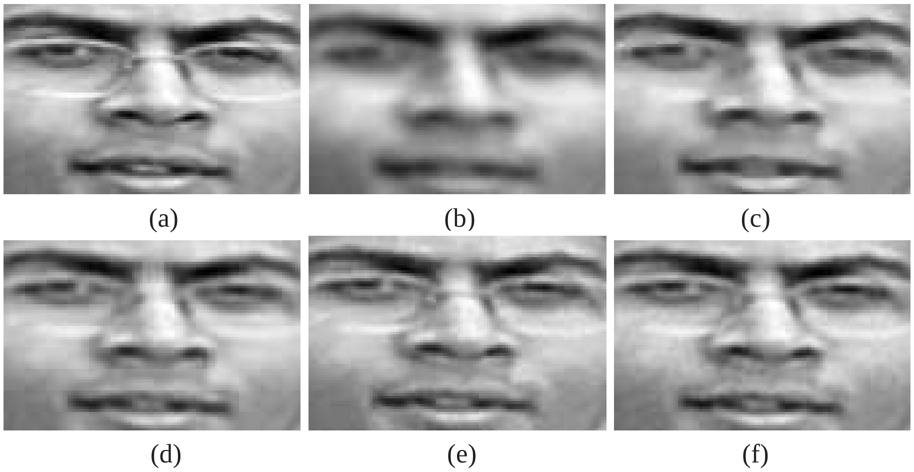


Fig. 7. Details of the image deconvolution experiment with a *face* image. (a) Original image. (b) Noisy blurred image. (c) Hyper-Laplacian [37] estimate (ISNR 5.16 dB). (d) Wavelet domain sparsity-based estimate [9] estimate (ISNR 6.1 dB). (e) LPA-ICI estimate [23] (ISNR 7.4 dB) (f) Parametric manifold-based estimate (ISNR 8.49 dB) suggested in this paper.

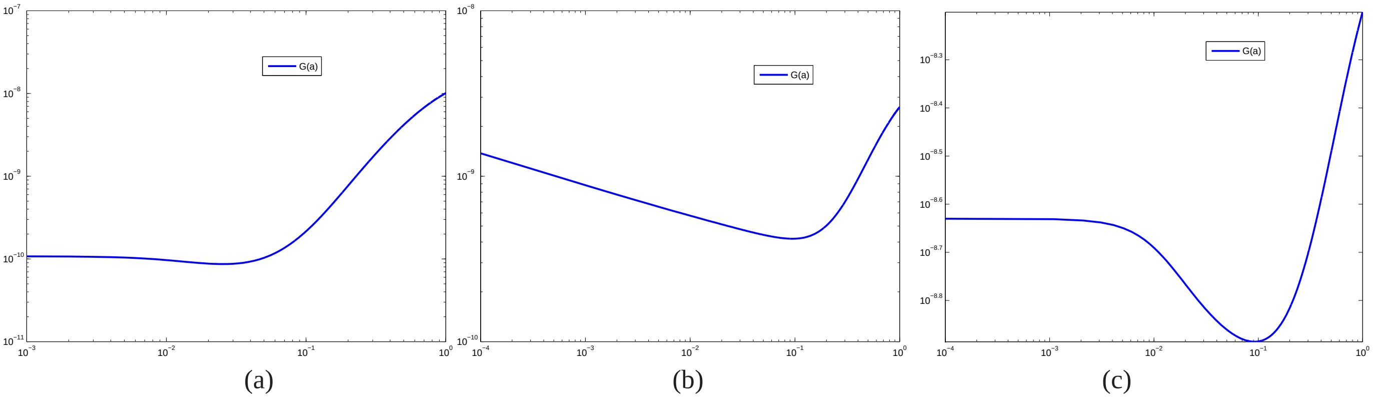


Fig. 8. GCV function for regularization with manifold prior. (a) *Barbara* Experiment. (b) *Koala* Experiment. (c) *flowers* Experiment.

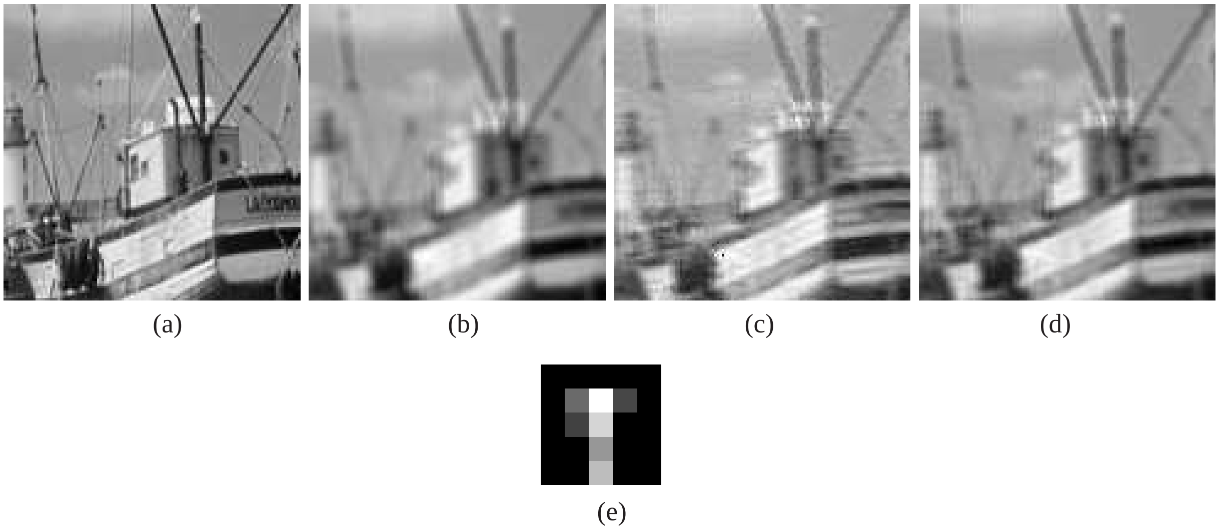


Fig. 11. Details of the blind deconvolution experiment with a *Boat* image. (a) Original image. (b) Blurred noisy image. (c) Result obtained by applying a blind deconvolution method in [39] (ISNR -0.19 dB). (d) Result obtained by applying parametric manifold deconvolution method using blur kernel estimated from [39] (ISNR 1.59 dB). (e) Estimated kernel.

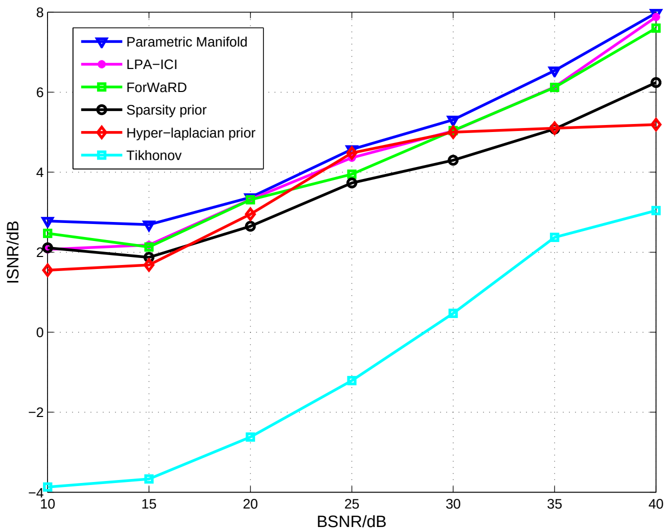


Fig. 9. ISNR performance of manifold-based algorithm compared to other methods as a function of BSNR

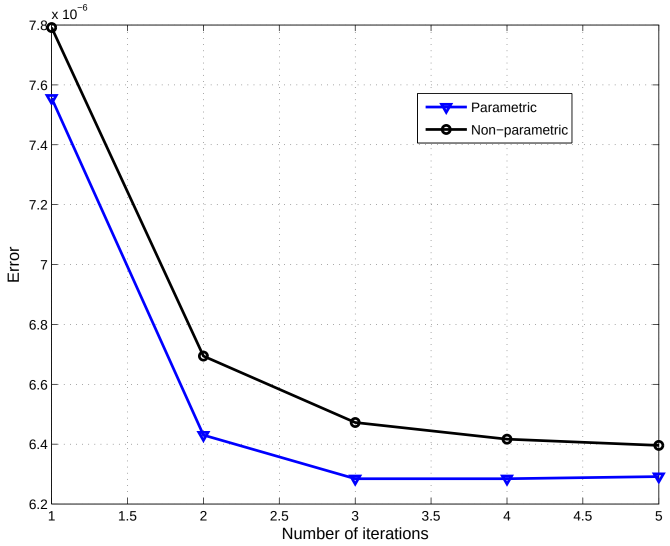


Fig. 10. $\frac{1}{MN}\|\mathbf{y}-\mathbf{H}\mathbf{x}^{(k)}\|_2^2$ vs. number of iterations to determine the stopping criteria.

References

[1] J. Ni, P. Turaga, V. M. Patel, and R. Chellappa, "Object dependent manifold priors for image deconvolution," in *Digital Image Processing and Analysis (DIPA)*, 2010.

[2] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*. New York, USA: Winston, 1977.

[3] J. Biemond, R. Lagendijk, and R. Mersereau, "Iterative methods for image deblurring," *Proceedings of the IEEE*, vol. 78, no. 5, pp. 856 –883, may 1990.

[4] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[5] J. Kalifa, S. Mallat, and B. Roug, "Deconvolution by thresholding in mirror wavelet bases," *IEEE Trans. on Image Processing*, vol. 12, no. 4, pp. 446–457, 2003.

[6] R. Neelamani, H. Choi, and R. G. Baraniuk, "Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems," *IEEE Trans. on Image Processing*, vol. 52, no. 2, pp. 418–433, 2004.

[7] D. L. Donoho, "Nonlinear solution of linear inverse problems by waveletvaguelette decomposition," *Applied and Computational Harmonic Analysis*, vol. 2, pp. 101–126, 1995.

[8] V. M. Patel, G. R. Easley, and D. M. Healy, "Shearlet-based deconvolution," *IEEE Trans. on Image Processing*, vol. 18, no. 12, pp. 2673–2685, 2009.

[9] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing:application to compressed sensing and other inverse problems*, vol. 1, no. 4, pp. 586–598, 2007.

[10] M. Figueiredo and R. Nowak, "An em algorithm for wavelet-based image restoration," *Image Processing, IEEE Transactions on*, vol. 12, no. 8, pp. 906 – 916, aug. 2003.

[11] J.-L. Starck, M. K. Nguyen, and F. Murtagh, "Wavelets and curvelets for image deconvolution: a combined approach," *Signal Process.*, vol. 83, no. 10, pp. 2279–2283, 2003.

[12] E. J. Candès and D. L. Donoho, "Curvelets - a surprisingly effective nonadaptive representation for objects with edges," *in Curves and Surface Fitting, A. Cohen, C. Rabut, and L. L. Schumaker, Eds. Nashville, TN: Vanderbilt Univ. Press*, 1999.

[13] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, Dec. 2005.

[14] A. L. Cunha, J. Zhou, and M. N. Do, "The nonsubsampled contourlet transform: Theory, design, and applications," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 3089–3101, Oct. 2006.

[15] G. R. Easley, D. Labate, and W. Q. Lim, "Sparse directional image representations using the discrete shearlet transform," *Appl. Comput. Harmon. Anal.*, vol. 25, no. 1, pp. 25–46, July 2008.

[16] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *Image Processing, IEEE Transactions on*, vol. 15, no. 2, pp. 3736–3745, Dec 2006.

[17] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *Image Processing, IEEE Transactions on*, vol. 17, no. 1, pp. 53–69, Jan 2008.

[18] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of IEEE*, vol. 98, no. 6, pp. 1045 – 1057, June 2010.

[19] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based superresolution," *IEEE Comput. Graph. Appl.*, vol. 22, pp. 56–65, March 2002.

[20] "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259 – 268, 1992.

[21] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Img. Sci.*, vol. 1, no. 3, pp. 248–272, 2008.

[22] L. He, A. Marquina, and S. Osher, "Blind deconvolution using tv regularization and bregman iteration," *International Journal of Imaging Systems and Technology*, vol. 15, no. 1, pp. 74–83, 2005.

[23] V. Katkovnik, K. Egiazarian, and J. Astola, "A spatially adaptive nonparametric regression image deblurring," *Image Processing, IEEE Transactions on*, vol. 14, no. 10, pp. 1469 –1478, oct. 2005.

[24] H. Takeda, S. Farsiu, and P. Milanfar, "Deblurring using regularized locally adaptive kernel regression," *Image Processing, IEEE Transactions on*, vol. 17, no. 4, pp. 550 –563, april 2008.

[25] G. Peyre, "Manifold models for signals and images," *Computer Vision and Image Understanding*, vol. 13, no. 2, pp. 249–260, 2009.

[26] P. N. Belhumeur and D. J. Kriegman, "What is the set of images of an object under all possible lighting conditions?" in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1996.

[27] d.-S. V. Tenenbaum, J. B. and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, December 2000.

[28] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, December 2000.

[29] A. Talwalkar, S. Kumar, and H. A. Rowley, "Large-scale manifold learning," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[30] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2006.

[31] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the 25th Very Large Database(VLDB) Conference*, 1999.

[32] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the Symposium on Computational geometry*, 2004, pp. 253–262.

[33] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet, "Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering," in *Neural Information Processing Systems (NIPS)*, 2003.

[34] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[35] N. Galatsanos and A. Katsaggelos, "Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation," *Image Processing, IEEE Transactions on*, vol. 1, no. 3, pp. 322 –336, jul 1992.

[36] P. C. Hansen, *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1998.

[37] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," in *Neural Information Processing Systems*, Vancouver, 2009.

[38] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluation blind deconvolution algorithms," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[39] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. Freeman, "Removing camera shake from a single photograph," *ACM Transactions on Graphics, SIGGRAPH 2006 Conference Proceedings, Boston, MA*, vol. 25, pp. 787–794, 2006.

PLACE
PHOTO
HERE

Jie Ni received the B.S. degree in automation from Tsinghua University, China in 2008. She is currently pursuing her PhD degree in the Department of Electrical and Computer Engineering at University of Maryland, College Park. Her research interests include computer vision, image processing and machine learning.

PLACE
PHOTO
HERE

Pavan Turaga (S05, M 09) is a Research Associate at the Center for Automation Research, University of Maryland, College Park. He received the B.Tech. degree in electronics and communication engineering from the Indian Institute of Technology Guwahati, India, in 2004, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park in 2008 and 2009 respectively. His research interests are in statistics and machine learning with applications to computer vision and pattern analysis. His published works include human activity analysis from videos, video summarization, dynamic scene analysis, and statistical inference on manifolds for these applications. He was awarded the Distinguished Dissertation Fellowship in 2009. He was selected to participate in the Emerging Leaders in Multimedia Workshop by IBM, New York, in 2008.

PLACE
PHOTO
HERE

Vishal M. Patel (M'01) is an ORAU postdoctoral fellow at the U.S. Army Research Laboratory. He received the B.S. degrees in electrical engineering and applied mathematics (with honors), the M.S. degree in applied mathematics from North Carolina State University, Raleigh, NC, in 2004 and 2005, respectively, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, in 2010.

His research interests include biometrics, radar imaging, inverse problems and pattern recognition. He is a member of Eta Kappa Nu, Pi Mu Epsilon and Phi Beta Kappa.

PLACE
PHOTO
HERE

Prof. Rama Chellappa received the B.E. (Hons.) degree from University of Madras, India, in 1975 and the M.E. (Distinction) degree from the Indian Institute of Science, Bangalore, in 1977. He received the M.S.E.E. and Ph.D. Degrees in Electrical Engineering from Purdue University, West Lafayette, IN, in 1978 and 1981 respectively. Since 1991, he has been a Professor of Electrical Engineering and an affiliate Professor of Computer Science at University of Maryland, College Park. He is also affiliated with the Center for Automation Research (Director) and the Institute for Advanced Computer Studies (Permanent Member). In 2005, he was named a Minta Martin Professor of Engineering. Prior to joining the University of Maryland, he was an Assistant (1981-1986) and Associate Professor (1986-1991) and Director of the Signal and Image Processing Institute (1988-1990) at University of Southern California, Los Angeles. Over the last 30 years, he has published numerous book chapters, peer-reviewed journal and conference papers. He has co-authored and co-edited books on MRFs, face and gait recognition and collected works on image processing and analysis. His current research interests are face and gait analysis, markerless motion capture, 3D modeling from video, image and video-based recognition and exploitation, compressive sensing, and hyper spectral processing.

Prof. Chellappa has received several awards, including an NSF Presidential Young Investigator Award, four IBM Faculty Development Awards, an Excellence in Teaching Award from the School of Engineering at USC, and two paper awards from the International Association of Pattern Recognition. He received the Society, Technical Achievement and Meritorious Service Awards from the IEEE Signal Processing Society. He also received the Technical Achievement and Meritorious Service Awards from the IEEE Computer Society. At University of Maryland, he was elected as a Distinguished Faculty Research Fellow, as a Distinguished Scholar-Teacher, received the Outstanding Faculty Research Award from the College of Engineering, an Outstanding Innovator Award from the Office of Technology Commercialization and an Outstanding GEMSTONE Mentor Award. In 2010, he was recognized as an Outstanding ECE by Purdue University. He is a Fellow of the IEEE, the International Association for Pattern Recognition and the Optical Society of America.

Prof. Chellappa served as the associate editor of four IEEE Transactions, as a Co-Editor-in-Chief of Graphical Models and Image Processing and as the Editor-in-Chief of IEEE Transactions on Pattern Analysis and Machine Intelligence. He served as a member of the IEEE Signal Processing Society Board of Governors and as its Vice President of Awards and Membership. He has served as a General and Technical Program Chair for several IEEE international and national conferences and workshops. He is a Golden Core Member of the IEEE Computer Society and served a two-year term as a Distinguished Lecturer of the IEEE Signal Processing Society. Recently, he completed a two-year term as the President of IEEE Biometrics Council.