Locally linear denoising on image manifolds

Dian Gong
Dept. of Electrical Engineering
Univ. of Southern California
Los Angeles, CA 90089

Fei Sha
Dept. of Computer Science
Univ. of Southern California
Los Angeles, CA 90089

Gérard Medioni
Dept. of Computer Science
Univ. of Southern California
Los Angeles, CA 90089

Abstract

We study the problem of image denoising where images are assumed to be samples from low dimensional (sub)manifolds. We propose the algorithm of locally linear denoising. The algorithm approximates manifolds with locally linear patches by constructing nearest neighbor graphs. Each image is then locally denoised within its neighborhoods. A global optimal denoising result is then identified by aligning those local estimates. The algorithm has a closed-form solution that is efficient to compute. We evaluated and compared the algorithm to alternative methods on two image data sets. We demonstrated the effectiveness of the proposed algorithm, which yields visually appealing denoising results, incurs smaller reconstruction errors and results in lower error rates when the denoised data are used in supervised learning tasks.

1 INTRODUCTION

Many algorithms developed for tasks in computer vision, such as object recognition, segmentation and others, assume that the input images contain little or no noise. Thus, for vision systems accomplishing those tasks, it is important to remove excessive noise in images at processing stages as early as possible. Image denoising is an important preprocessing step for achieving that goal (Elad and Aharon, 2006; Buades et al., 2005; Perona and Malik, 1990). Denoising techniques are also widely used in computer graphics (Fleishman et al., 2003), digital photography (Fergus et al., 2006) and other applications.

Principal component analysis (PCA) is a popular denoising technique. It is especially effective when images are contaminated with small amounts of Gaussian noise. Probabilistic approaches, based on learning priors for appearance, geometry and other visually salient properties, have also been intensively studied (Roth and Black, 2005; Kivinen et al., 2007). In these works, the prior models are learnt using image corpora, such as those of natural scenes, where images are randomly collected and are not meaningfully related to each other (Martin et al., 2001). Many state-of-the-art denoising techniques are based on statistical signal processing and optimal filtering (Elad and Aharon, 2006; Buades et al., 2005; Guerrero-Colon et al., 2008; Dabov et al., 2007; Portilla et al., 2003). A key assumption in many of these work is that for a given image, many pixels' neighborhoods (or local patches) are similar to each other. Such similarity is leveraged to estimate models of noise and clean images.

While the majority of existing work has been focusing on denoising a single image, we investigate the problem of denoising collectively a collection of images. In many cases, latent intrinsic structures underpin those images. For instance, an image library of an object can be compactly described with a few parameters such as the lighting condition, the camera position, etc. We assume that these latent variables lie on a smooth low dimensional manifold. Identifying image manifolds is an active research topic in manifold learning and latent variable models (Tenenbaum et al., 2000; Roweis and Saul, 2000; Belkin and Niyogi, 2003; Lawrence, 2005). We consider the problem of denoising in this context. Specifically, we view images as random samples (with noise) from the manifold. A natural question arises: can the intrinsic structure be exploited for denoising? Note that the intrinsic structure is often unknown a priori, therefore needs to be inferred from the (noisy) data. How can we achieve robust denoising and inference at the same time?

Our work investigates these questions. We propose a simple and effective procedure for denoising data on

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9. Copyright 2010 by the authors.

manifolds. Our study shows clearly that exploiting the intrinsic structure of image collections is advantageous. Our iterative procedure consists of 3 steps: i) construct a nearest neighbor graph to approximate the manifold with locally linear patches; ii) denoise data points locally within each patch; iii) align denoised data globally with regularization enforcing smoothness on manifolds. Each of the three steps is computationally tractable, involving nearest neighbor search, matrix eigendecomposition and matrix inversion. We applied our algorithm to image denoising on manifolds of handwritten digits and faces. We evaluate the quality of the denoising by visual inspection, deviation from uncorrupted images and classification error rates on denoised images. We compare our algorithm to alternative methods systematically under various types of noise conditions. Our algorithm generally outperforms other approaches.

The rest of the paper is organized as follows. In section 2, we summarize briefly related work. We derive and describe our algorithm in section 3. Experimental evaluation is presented in section 4. We discuss future research directions in section 5.

2 RELATED WORK

Manifold learning algorithms also aim to exploit intrinsic structures in data. They are different from our effort in their primary goals of discovering and projecting data onto low dimensional structures for visualization and exploratory data analysis (Tenenbaum et al., 2000; Roweis and Saul, 2000; Belkin and Niyogi, 2003). In contrast, the primary goal of denoising is to obtain denoised output in the same dimensionality as the noisy input. Note that, for manifold learning algorithms, it is possible to build statistical models or functions that map data in the low dimensional space to the original input space (Lawrence, 2005; Teh and Roweis, 2003; Gao et al., 2008). The outputs of those models could be seen as denoised inputs. Intuitively, it is difficult to ensure the effectiveness and advantage of this type of denoising procedures since both phases – projection and backward mapping – introduce errors. Empirically, our experimental results did not support this procedure as a robust option for denoising.

Our work is more similar in spirit to the diffusion map based denoising algorithm (Hein and Maier, 2007). They view denoising as reversing a diffusion process of which graph Laplacian is the generator. Both their and our approaches use graph Laplacian for regularization. However, our approach is significantly different from theirs as our iterative procedure uses intermediate denoising outputs differently to refine the existing solution. In particular, their approach tends to overly smooth the inputs. Our approach is less sensitive to that problem.

3 LOCALLY LINEAR DENOISING

We assume that the (image) data lies on a d-dimensional smooth submanifold embedded in an ambient space of dimensionality $D > d$. Let $\{\boldsymbol{x}_i \in \Re^D, i = 1, 2, \ldots, N\}$ be N data points sampled from the manifold. Let $\boldsymbol{X} \in \Re^{D \times N}$ denote the matrix where $\boldsymbol{x}_i$ is the i-th column. Let $z_i \in \Re^d$ denote the corresponding coordinates for $\boldsymbol{x}_i$ in the low dimensional space. We assume that there exists a smooth function f, mapping the low dimensional coordinates to the high dimensional space: $\boldsymbol{x}_i = f(\boldsymbol{z}_i) + \epsilon_i$, where ϵ_i represents noise.

We denote $f(\boldsymbol{z}_i)$ by $\boldsymbol{y}_i$, ie, the *noiseless* data. We define the shorthand notation $\boldsymbol{Y}$ as we did for $\boldsymbol{X}$. We are interested in denoising noisy data $\boldsymbol{x}_i$ thus identifying $\boldsymbol{y}_i$. Note that our goal is different from most manifold learning algorithms, which aim to identify $\{\boldsymbol{z}_i\}$ and sometimes the function $f(\cdot)$ (Tenenbaum et al., 2000; Roweis and Saul, 2000; Belkin and Niyogi, 2003; Zhang and Zha, 2004).

In what follows, we start by describing and deriving our algorithm in details. We then analyze the algorithm briefly and discuss a few possible extensions.

3.1 The LLD ALGORITHM

Our approach hinges on the basic notion that a manifold can be seen as a collection of overlapping small linear patches. Moreover, if data are sampled densely, these small patches can be approximated with nearest neighborhoods in point clouds. This is the same intuition behind many manifold learning algorithms (Tenenbaum et al., 2000; Roweis and Saul, 2000). Our algorithm exploits this intuition by denoising in each patch independently. Then we compute a global assignment of denoising outputs that align with local results as much as possible. Specifically, our algorithm consists of the following three steps:

3.1.1 Constructing nearest neighbor graph

We construct a weighted nearest neighbor graph from the sampled data points. Let K denote the number of nearest neighbors for each data point (including itself) and $\mathcal{N}_i$ denote the neighbors of $\boldsymbol{x}_i$. We use w_{ij} to denote the weight of the edge between the samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Many choices of w_{ij} are possible. In this work, we have chosen the commonly used Gaussian kernels $w_{ij} = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/\sigma^2)$ if $\boldsymbol{x}_j \in \mathcal{N}_i$ or $\boldsymbol{x}_i \in \mathcal{N}_j$, and 0 otherwise. Let $\boldsymbol{W}$ denote the weight matrix where the elements are w_{ij}.

3.1.2 Denoising locally

We view $\boldsymbol{x}_i$ and K points in its neighborhoood $\mathcal{N}_i$ as random samples from a linear subspace, approximating the manifold around the point $\boldsymbol{x}_i$. We denoise these K points with principal component analysis (PCA) (other strategies are also possible). Let $\boldsymbol{X}_i$ denote the points in $\mathcal{N}_i$. The *local* estimate of $\boldsymbol{y}_i$ — denoised $\boldsymbol{x}_i$ — is given by the reconstruction of $\boldsymbol{x}_i$ with the d_i principal components of $\boldsymbol{X}_i$. Similarly, we compute reconstructions for other points in $\mathcal{N}_i$. Concretely, let $\boldsymbol{U}_i \in \Re^{D\times d_i}$ denote the d_i principal components. The reconstruction $\boldsymbol{Q}_i \in \Re^{D\times K}$ for these points is given by

$$\boldsymbol{Q}_i = \boldsymbol{U}_i\boldsymbol{U}_i{}^{\mathrm{T}}\boldsymbol{X}_i\left(\boldsymbol{I} - \boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}/K\right) + \boldsymbol{X}_i\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}/K \qquad (1)$$

where $\boldsymbol{e}$ is a length-K column vector whose element values are ones and $\boldsymbol{I}$ is the identity matrix.

The number of the principal components d_i can be estimated adaptively, mindful of the inhomogeneous distribution of the noise at different parts of the manifold. This is in sharp contrast to many manifold learning algorithms where a *global* dimensionality d needs to be estimated. When there is noise in the data, estimating a global d is challenging as the noise and the curvature of the manifold interplay. In our work, d_i is estimated with simple methods such as thresholding on the residual variances.

3.1.3 Aligning globally

In addition to having its own neighborhood, any point $\boldsymbol{x}_i$ can be in the neighborhoods of other data points as the approximating linear subspaces overlap for a densely sampled manifold. Intuitively, the local denoising results for $\boldsymbol{x}_i$ from other neighborhoods reflect also information about the "true" location $\boldsymbol{y}_i$. To integrate this information from every neighborhood, we seek a global assignment of denoising result $\boldsymbol{Y}$ that minimizes the sum of discrepancies to all neighborhoods. Specifically, we minimize the following loss function

$$\mathcal{L}_A = \sum_i \sum_{j:\ i\in\mathcal{N}_j} \|\boldsymbol{y}_i - \boldsymbol{Q}_j(i)\|_2^2 \qquad (2)$$

where $\boldsymbol{Q}_j(i)$ stands for the local denoising result for $\boldsymbol{x}_i$ from the neighborhood of $\boldsymbol{x}_j$.

The loss function $\mathcal{L}_A$ can be expressed compactly with data selection matrices (Zhang and Zha, 2004). Let $\boldsymbol{S} = [\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_N]$ be a $1 \times N$ block matrix where each block $\boldsymbol{S}_i \in \Re^{N\times K}$ corresponds to the neighborhood of $\boldsymbol{x}_i$. Moreoever, $\boldsymbol{S}_i$ is a binary matrix and its element $s_{jk} = 1$ if and only if $\boldsymbol{x}_j$ is the k-th nearest neighbor of $\boldsymbol{x}_i$. Note that $\boldsymbol{X}\boldsymbol{S}_i = \boldsymbol{X}_i$. The loss function $\mathcal{L}_A$ is expressed as

$$\mathcal{L}_A = \|\boldsymbol{Y}\boldsymbol{S} - \boldsymbol{Q}\|_F^2 \qquad (3)$$

where $\boldsymbol{Q} = [\boldsymbol{Q}_1\ \boldsymbol{Q}_2\ \cdots\ \boldsymbol{Q}_N]$ and the subscript F indicates the Frobenious norm.

In addition to aligning the global coordinates $\boldsymbol{Y}$ with the local estimates $\boldsymbol{Q}$, we also seek an output that is smooth on the manifold. To this end, we also minimize the total variation of $\boldsymbol{Y}$ on the graph (Hein and Maier, 2007). The total variation is computed as the squared norm of the (discrete) difference, $\mathcal{L}_G = \sum_{m=1}^{D}\|\nabla\boldsymbol{Y}_m\|^2$, where $\boldsymbol{Y}_m$ is the m-th row of the denoised data $\boldsymbol{Y}$ and ∇ is the discrete difference, approximating the gradient on continuous manifold (Chung, 1997). The loss can be written in terms of the graph Laplacian, analogous to the Laplacian-Bertrami operator on smooth manifolds,

$$\mathcal{L}_G = \sum_{m=1}^{D}\boldsymbol{Y}_m\boldsymbol{L}\boldsymbol{Y}_m{}^{\mathrm{T}} = \mathrm{trace}(\boldsymbol{L}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}) \qquad (4)$$

The graph Laplacian $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{W}$ is defined in terms of the weight matrix $\boldsymbol{W}$ for the graph and the diagonal matrix $\boldsymbol{D}$ with the diagonal element of $D_{ii} = \sum_j w_{ij}$.

The loss function $\mathcal{L}_G$ attains its minimum at a constant $\boldsymbol{Y}$, which would result in a large loss for the alignment $\mathcal{L}_A$. To tradeoff, we adopt the same regularization framework proposed in (Hein and Maier, 2007) and compute the optimal solution to the following loss

$$\mathcal{L}(\lambda) = \|\boldsymbol{Y}\boldsymbol{S} - \boldsymbol{Q}\|_F^2 + \lambda\,\mathrm{trace}(\boldsymbol{L}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}) \qquad (5)$$

as the denoising result: $\boldsymbol{Y}^* = \arg\min_{\boldsymbol{Y}}\mathcal{L}(\lambda)$. Empiricall, the coefficient $\lambda \geq 0$ is chosen with validation sets. Note that the optimal denoising result has a closed-form solution given by

$$\boldsymbol{Y}^*(\lambda) = \boldsymbol{Q}\boldsymbol{S}^{\mathrm{T}}(\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}} + \lambda\boldsymbol{L})^{-1} \qquad (6)$$

The matrix product $\boldsymbol{\Lambda} = \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}$ is a diagonal matrix. Specifically, its diagonal element Λ_{ii} is the number of nearest neighborhoods that $\boldsymbol{x}_i$ belongs to.

The algorithm listing in Fig. 1 reviews the key steps of our algorithm. Note that our algorithm depends on the nearest neighbor graph which is estimated from the (noisy) data samples $\boldsymbol{X}$. After computing the denoised output $\boldsymbol{Y}^*(\lambda)$ in eq. (6), the graph can be re-estimated with denoised outputs. The denoising can then be iteratively refined on the new graph. While a formal proof of convergence of this iterative process is left for future work, we observe convergence after a few iterations to a stable solution in our experiments.

```
Input $X$: noisy data, $K$: number of nearest neighbors, $\lambda$: regularization coefficient for graph Laplacian
    S1 construct $K$ weighted nearest neighbor graph
    S2 denoise locally with eq. (1) to compute $Q$
    S3 align globally with eq. (6)
    S4 (optionally) iterate from S1, use the results in S3 in place of $X$
Output $Y^*$: denoising results
```

Figure 1: Locally linear denoising (LLD) algorithm

3.2 ANALYSIS

In the following, we analyze the LLD algorithm briefly and contrast to a closely related approach (Hein and Maier, 2007). To gain intuition, we first consider the case when $\lambda = 0$. Note that, when λ approaches ∞, as mentioned before, the solution $\boldsymbol{Y}^*(\lambda)$ of eq. (6) becomes trivially constant so that the solution is infinitely smooth.

On the other extreme, however, when λ is zero, there is no enforcement on the smoothness of the denoised output. Instead, the solution takes the simple form of $\boldsymbol{Y}^*(0) = \boldsymbol{Q}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}$. A short calculation reveals that $\boldsymbol{Y}^*(0)$ is the *average* of all local estimates. It is interesting to note that this simple procedure works well in some denoising tasks, as evidenced by empirical study in later sections. Also, even for $\lambda = 0$, due to the iterative nature of the LLD algorithm, averaging changes from one iteration to the other as the nearest neighbors are recomputed every iteration. Furthermore, if $\boldsymbol{Q}$ is computed with nonlinear methods from $\boldsymbol{X}$ (as opposed to linear projections with PCA), the overall averaging effect is highly nonlinear and compounded.

When $\lambda \ll 1$, we can approximate the matrix inverse in eq. (6) with $(\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}+\lambda\boldsymbol{L})^{-1} \approx \boldsymbol{\Lambda}^{-1}(\boldsymbol{I} - \lambda\boldsymbol{L}\boldsymbol{\Lambda}^{-1})$ (by applying Taylor expansion). The solution is then approximated with

$$\boldsymbol{Y}^*(\lambda) \approx \boldsymbol{Y}^*(0)(\boldsymbol{I} - \lambda\boldsymbol{\Lambda}^{-1}) + \lambda\boldsymbol{Y}^*(0)\boldsymbol{D}^{-1}\boldsymbol{W}\boldsymbol{\Lambda}^{-1} \quad (7)$$

The first term approximately takes the form of discounted simple averages of local estimates. The second term reveals more insight. Specifically, the local estimate $\boldsymbol{q}_j$ also contributes to the global assignment $\boldsymbol{y}_i$ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are connected in the weighted nearest neighbor graph. Moreover, for the Gaussian kernel we have taken to compute the weights, the contribution is positively proportional to the weight w_{ij}. Intuitively, if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are close to each other, their local estimates should be similar to each other. Hence, information from the local estimate $\boldsymbol{q}_j$ could be used for estimating $\boldsymbol{y}_i$.

Hein et al has recently proposed a diffusion map based denoising algorithm (Hein and Maier, 2007). Similar to ours, their algorithm computes directly the high dimensional denoising outputs and incorporates graph Lapalacian as a regularization to favor smooth solutions. Their iterative procedures take the form (in the notation of this paper)

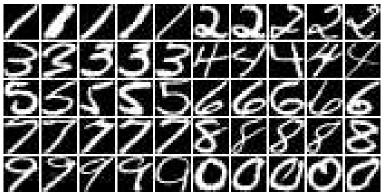

Figure 2: Examples of clean USPS digit images

$$\boldsymbol{Y} \leftarrow \boldsymbol{Y}(\boldsymbol{I} + \lambda\boldsymbol{L})^{-1} \quad (8)$$

While the two updates in eq. (6) and eq. (8) appear similar, key differences exist. In eq. (8), the updated denoising result $\boldsymbol{Y}$ (on the left side) is used directly in the right side to be refined. In eq. (6), the updated denoising result affects the refined output at the next time step *indirectly* through the computation of *local estimates* $\boldsymbol{Q}$ on the right side. Note that the computation of $\boldsymbol{Q}$ requires recomputing the nearest neighbor graph as well as recomputing projection matrix (cf. eq. (1)). Therefore, the two denoising algorithms are unlikely to converge to the same stationary point.

We gain further insight by inspecting again the special case when $\lambda = 0$. Note that eq. (8) immediately reaches a fixed point for any $\boldsymbol{Y}$. For the LLD algorithm eq. (6), this is not necessarily true. In the next section, we compare the LLD algorithm to their algorithm empirically and discover significant differences in applications.

4 EXPERIMENTS

4.1 EVALUATION METHODOLOGY

We evaluate the performance of the locally linear denoising (LLD) algorithm on two data sets: a subset of the USPS handwritten digit images, which contains 200 images per digit class, and the ORL face images with resolution reduced from 112 × 92 to 28 × 23. We chose them for their different characteristics in number of samples, dimensionality, and intrinsic structures. We compare the performance of our approach to that of four denoising algorithms: PCA, the diffusion map based manifold denoising (DM-MD) algorithm (Hein and Maier, 2007), Non-Local Means (NLM) and BLS-GSM (Buades et al., 2005; Portilla et al., 2003; Guerrero-Colon et al., 2008). The later two methods are designed to denoise one image at a

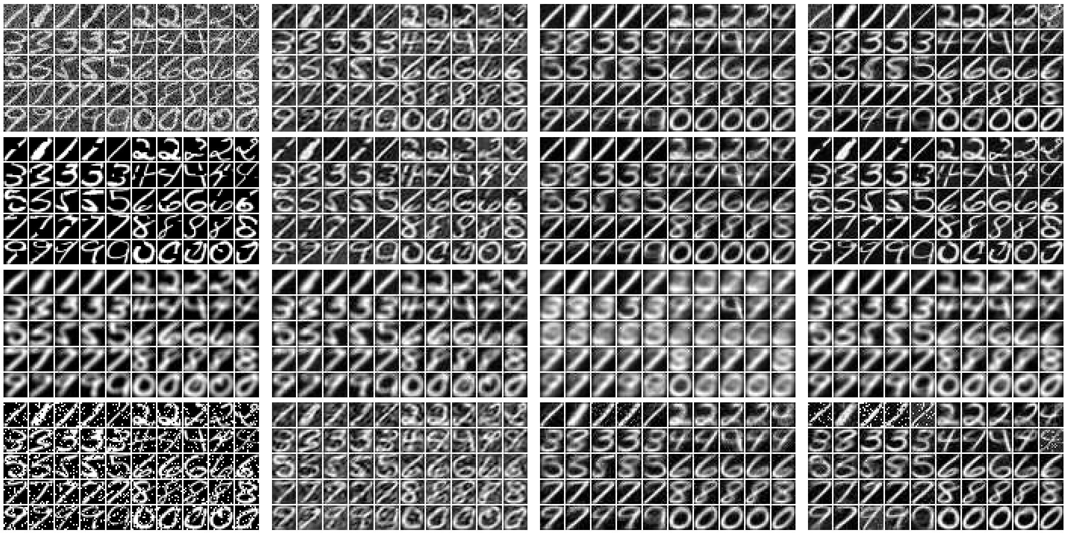

Figure 3: Denoising by PCA and manifold based algorithms on USPS data. Top row to bottom: different noise types (Gaussian, occlusion, motion blur and salt-and-pepper). Left column to right: no denoising, PCA, DM-MD (Hein and Maier, 2007) and LLD (cf. section 3) with $\lambda = 0$. Denoised images are visually more appealing. DM-MD tends to overly smooth while LLD does not perform well with occlusion noise. For DM-MD, the number of nearest neighbors are $K = 80, 80, 30, 100$ for each noise type respectively. For LLD, $K = 30$ for all noise types.

time, therefore, do not rely on intrinsic latent structures in image collections.

We also tried a two-step denoising strategy where we project to a low dimensional space first and then use a learned statistical model to map back to the original input space (Teh and Roweis, 2003). Preliminary results did not support this strategy as a viable option for robust denoising. One possible reason is that both steps introduce errors and there is no "global" criteria controlling the quality of the final denoised images. We omit those results. Note that, the LLD algorithm computes denoised images without identifying low dimensional embeddings.

We investigate the robustness of denoising algorithms to different types of noise. We treat the original images as "clean" images and synthesize noisy images by adding noise to them: Gaussian noise imposed on the pixel intensities, random occlusion patches, motion blurring as caused by camera movements, and salt-and-pepper noise where each pixel's intensity is randomly flipped at a probability of 20% to its complement alue. Denoising algorithms that are resilient to different noise types are highly desirable in practice as inferring noise types is often challenging.

We examine the quality of denoising with visual inspection. We also apply two quantitative metrics: reconstruction errors between denoised outputs and the clean images; as well as classification errors on the denoised outputs with classifiers trained on clean data. Note that the denoising algorithms are not told which images are noisy ones. Therefore, all images are denoised. As a consequence, the original clean images will be contaminated while noisy images are cleaned. We report results separately on them. Fig. 2 and 6 show examples of clean images from these two data sets. We report findings on the USPS data first, followed by those on the ORL face images.

The LLD algorithm depends on the regularizer coefficient λ and the number of nearest neighbors K. We chose them based on cross-validation using either of the two quantitative metrics. Parameters for other algorithms are tuned similarly.

4.2 USPS DIGIT IMAGES

Denoising results Fig. 3 displays the denoising results on noisy images by 3 algorithms under the four types of noise. We added noise to 100 images (10 per class) and show 50 of them (results for the other 50 are similar). Overall, manifold based denoising algorithms yield more visually appealing reconstructions. PCA introduces suppositions of images from all classes as it adds to the reconstructed images with a mean image computed over the whole data set. The diffusion map based algorithm tends to overly smooth, also noted in (Hein and Maier, 2007). This is not necessarily a disadvantage: under the occlusion noise, this algorithm is the only one that can connect broken strokes caused by occlusion, thus more visually appealing.

In Table 1, we quantify the denoising quality in terms

Table 1: Reconstruction errors and misclassification rates (in percentage) by multiclass SVM classifiers on the USPS data. The error rates are shown inside the parentheses. Without denoising, corresponding to the column heads of "none", no reconstruction error is reported. In both measures, LLD outperforms other 2 methods in most cases. Parameters are set as the same as those in Fig. 3.

Noise	Noisy images				Original clean images			
type	none	PCA	DM-MD	LLD	none	PCA	DM-MD	LLD
Gaussian	- (20)	640 (19)	920 (11)	719 (13)	- (16)	518 (16)	918 (19)	665 (15)
Occlusion	- (18)	615 (18)	949(15)	524 (17)	- (16)	390 (16)	918 (20)	333 (15)
Blurring	- (20)	964 (20)	1228 (52)	1043 (22)	- (16)	420 (16)	906 (18)	274 (16)
Salt&Pepper	-(22)	785 (14)	977 (15)	860 (14)	- (16)	518(16)	938 (22)	666 (15)

of reconstruction errors, which is the sum of the squared differences in pixel intensities. As a reference, the amount of noise added to the images are 805, 520, 968 and 1137, respectively for each type of noise. On noisy images, PCA incurs the smallest amount of reconstruction errors and LLD has smaller errors than DM-MD. On the original clean images, PCA contaminates less in the cases of Gaussian and salt-and-pepper noise, while LLD contaminates less for occlusion and blur noise. DM-MD has the highest errors.

Classification An important use of denoising is to preprocess data for supervised learning tasks. In Table 1, we compare the misclassification rates using different denoising algorithms. The misclassification rates are numbers displayed inside parentheses. SVM classifiers were trained on 200 clean images (20 images per class) and we tested the classifiers with 100 noisy images and 1700 originally clean images. Almost all algorithms improve over the baseline without denoising. The DM-MD algorithm performs the best on noisy images with Gaussian and occlusion noise. However, it does so at the expense of increasing error rates on original clean images. The LLD algorithm attains the smallest error rates in most categories. In particular, it is able to achieve so by reducing error rates more than other algorithms on both noisy images and original clean images. The exception being on the motion bluring images, the error rate was increased slightly from 20% to 22%.

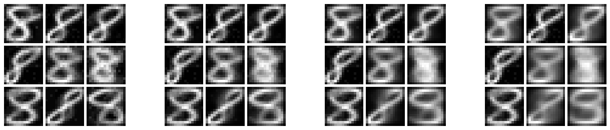

Figure 4: Denoising by LLD with different amount of regularization. λ is increased from left to right. Larger λ leads to oversmoothing, similar to the DM-MD algorithm.

Effects of graph regularization In the results we have reported so far, we have set the parameter λ to 0 in the LLD algorithm. A nonzero λ trades off the errors of alignment (see eq. (3)) and the smoothness of the denoised images. We experimented with different settings of λ under the salt-and-pepper noise. The optimal value for λ combined parameters K and d is chosen with the smallest classification error rates on validation data sets. Fig. 4 shows that a smaller λ retains the "grain" in the image while a larger one often oversmooths. The λ we used in this experiment are 0, 0.25, 1.5 and 9 respectively. Fig. 5 reports the classification performance of the LLD algorithm with two settings $\lambda = 0$ and $\lambda = 0.01$, as well as other algorithms. Note that while LLD($\lambda = 0$) achieves a better overall error rate than competitive methods, its error rates on noisy images were worse than that of DM-MD. With a small amount of regularization, the error rates of LLD($\lambda = 0.01$) on noisy images were significantly lower than both DM-MD and without regularization. Furthermore, the error rates on the original clean images were improved too, though not significantly. For this experiments, we have used 10 clean images per class to build a SVM classifier. On the clean images, the error rate is 30.4%. Therefore, an interesting observation is that both DM-MD and the two LLDs are able to improve the error rates from this baseline. We believe this is the benefits of semi-supervised learning,

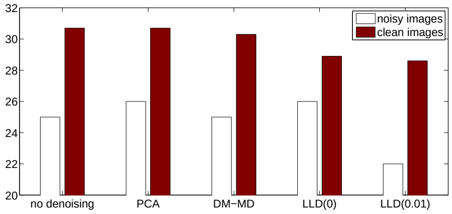

Figure 5: Misclassification rates (in percentage) of LLD with different regularization and other methods on USPS data with salt-and-pepper noise. Small regularization improves error rates on both noisy and original clean images (note that they are "denoised" too, effectively being introduced with noise). See text for details.

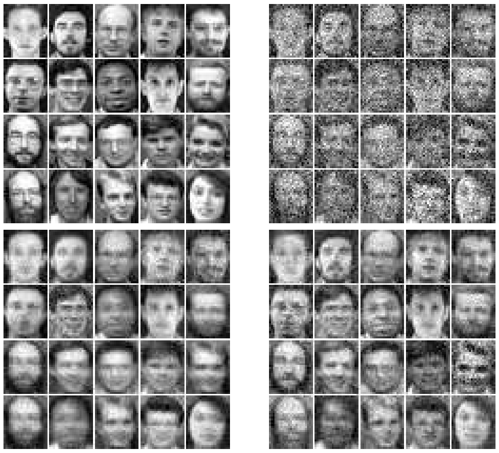

Figure 6: Visualization of denoised ORL images. From left column to right and top row to bottom: clean images, images with Gaussian noise, denoised with MD-DM and LLD algorithms respectively.

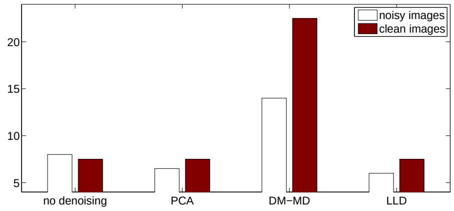

Figure 7: Misclassification rates (in percentage) for ORL data with various denoising algorithms (note that, "denoising" clean images introduces noises in effect). See text for details.

as discussed in (Hein and Maier, 2007).

4.3 ORL FACE IMAGES

The ORL face image data set has a total of 400 images from 40 subjects. We add noise to randomly chosen 200 images and retain the other 200 images as clean samples. We use 4 clean images per subject to train a 40-way classifier to distinguish subjects. Our test set for classification tasks contains 1 clean image and 5 noisy images per subject. The performance of two manifold based denoising algorithms under Gaussian noise are displayed in Fig. 6. We drew similar conclusions about these two algorithms as in our previous experiments on USPS data. In particular, we note that the DM-MD algorithm oversmooths, which would lead to inferior classification results. This is confirmed in Fig. 7. Note that using clean images only the classifier has a classification error of 5.8%. Furthermore, note that the results are obtained at choosing the number of nearest neighbors be 60 for LLD and 20 for DM-MD. Both numbers are greater than the number of images per subject in the data set. This indicates that the LLD algorithm is capable of using information from similar images, though not necessarily clustered in terms of subjects, to get rid of noise.

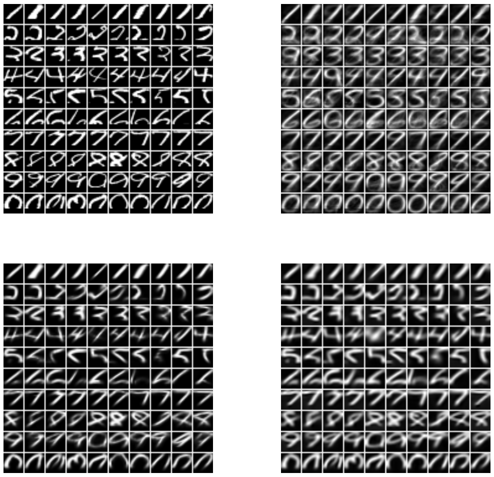

Figure 8: Comparison among various denoising algorithms. From top row to bottom and from left column to right: images of digits contaminated with narrow black bands at the bottom, denoising by the LLD algorithm, denoising by Non-Local Means, denoising by BLS-GSM. The LLD algorithm is able to recover partially occluded areas by the bands.

4.4 COMPARISON TO SINGLE-IMAGE DENOISING

The LLD algorithm *collectively* denoises all images. This is different from many existing approaches, which denoises one image at a time. A drawback of those approaches is that they cannot benefit from intrinsic structure, such as image manifolds.

To exemplify this, we compared the LLD algorithm to Non-Local Means (NLM) and BLS-GSM (Buades et al., 2005; Portilla et al., 2003; Guerrero-Colon et al., 2008), both regarded as state-of-the-art denoising algorithms. Specifically, we added a narrow black band to the bottom of 100 USPS digit images, as shown in Fig. 8. The LLD algorithm denoises these noisy images by using the rest clean images in the data set and is able to recover partially from the occlusion. On the other hand, both NLM and BLS-GSM cannot recognize the bands as noises.

If the intrinsic structure is contaminated by strong noise, the LLD algorithm is prone to extract information from wrong images to denoise. For instance, images of digit 2 with bottom bandnoises can be easily confused as images of digit 7. Therefore, LLD denoises by mixing both types of images. NLM and BLS-GSM are immune to this problem as these images are processed independently. Thus, it is interesting to explore whether and how we can combine the advantages of both types of techniques.

5 CONCLUSION

In this paper, we study the problem of image denoising when images lie on intrinsic low dimensional structures such as submanifolds. We propose locally linear denoising (LLD) to exploit such structures. The algorithm integrates local denoising results through a global alignment process that minimizes discrepancies in reconstruction with different local neighborhoods, balanced by graph Laplacian regularization to prefer smooth solutions on the manifolds. The algorithm is evaluated and compared to other state-of-the-art denoisubg methods. The results are encouraging: on both handwritten digit and face images, the proposed algorithm yields visually appealing denoising results, incurs small reconstruction errors and results in low error rates when the denoised data are used in supervised learning tasks.

We view the algorithm proposed in this paper as a general strategy. For example, the local denoising step can be easily adapted to other approaches that are more robust, for instance, robust PCA (Huber, 1981).

Field of Experts also learns a prior model of image patches from collection of images (Roth and Black, 2005). The prior does not depend on the contents of the data, while LLD does. It would be interesting to have a synergy between the two methods. We leave these opportunities to improve to future work.

Acknowledgments

This work is supported in part by NIH Grant EY01609 (Dian Gong and Gérard Medioni) and NSF Grant 0957742 (Fei Sha).

References

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, June 2003.

A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In *Proc. CVPR*, volume 2, pages 60–65, 2005.

F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 2nd edition edition, May 1997.

K. Dabov, A. Foi, V. Katkovnik, and K. O. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

M. Elad and M. Aharon. Image denoising via learned dictionaries and sparse representation. In *Proc. CVPR 2006*, volume 1, pages 895–900, 2006.

R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single image. *ACM Transactions on Graphics and SIGGRAPH*, 25:787–794, 2006.

S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. *ACM Transactions on Graphics and SIGGRAPH*, 22:950–953, July 2003.

Y. Gao, K. Chan, and W. Yau. Manifold denoising with Gaussian process latent variable models. In *Proceedings of Intl. Conf. on Pattern Recognition*, pages 1–4, 2008.

J. A. Guerrero-Colon, E. P. Simoncelli, and J. Portilla. Image denoising using mixtures of Gaussian scale mixtures. In *ICIP*, pages 565–568. IEEE, 2008.

M. Hein and M. Maier. Manifold denoising. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 561–568. MIT Press, Cambridge, MA, 2007.

P. J. Huber. *Robust Statistics*. Wiley-Interscience, February 1981.

J. J. Kivinen, E. B. Sudderth, and M. Jordan. Image denoising with nonparametric hidden markov trees. In *Proc. ICIP 2007*, volume 3, pages 121–124, 2007.

N. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *J. of Machine Learning Research*, 6:1783–1816, November 2005.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on PAMI*, 12: 629–639, July 1990.

J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, 2003.

S. Roth and M. J. Black. Fields of experts: a framework for learning image priors. In *Proc. CVPR 2005*, volume 2, pages 860–867, 2005.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, December 2000.

Y. W. Teh and S. Roweis. Automatic alignment of local representations. In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems 15*, pages 841–848. MIT Press, Cambridge, MA, 2003.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, December 2000.

Z. Zhang and H. Zha. Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *SIAM J. of Scientific Computing*, 26:313–338, 2004.