Learning Deformations

Anonymous ECCV submission

Paper ID 385

Abstract. Many vision problems, such as object recognition and image synthesis, are greatly impacted by deformation of objects. In this paper, we develop a deformation model based on Lie algebraic analysis. This work aims to provide a generative model that explicitly decouples deformation from appearance, which is fundamentally different from the prior work that focuses on deformation-resilient features or metrics. Specifically, the deformation group for each object can be characterized by a set of Lie algebraic basis. Such basis for different objects are related via parallel transport. Exploiting the parallel transport relations, we formulate an optimization problem, and derive an algorithm that jointly estimates the deformation basis for a class of objects, given a set of images resulted from the action of the deformations. We test the proposed model empirically on both character recognition and face synthesis.

1 Introduction

The changes in shapes of objects, often referred to as *deformations*, are widely observed in computer vision data. In many problems, particularly object recognition based on appearance, the performance can be greatly influenced by deformations. Whereas the past decades have seen tremendous efforts devoted to the development of features and classifiers that are resilient to variations of shapes and poses, the modeling of deformations has been not been extensively explored. In this paper, we focus on modeling deformations, aiming to develop a method that can decouple deformation from appearance of an observed image.

In past decades, a variety of approaches have been developed to address the issue of deformations, for which we provide a brief review in next section. Careful examination of previous shows that they are limited in several aspects: (1) While extensive research [**?**,**?**] has been performed on image manifold modeling, this does not lead to effective modeling of deformations. The problem here is that the differences between neighboring images are due to the compound effects of deformations and other contributing factors, and these approaches lack a mechanism to decouple the effects. (2) The methods for deformation-resilient metrics [**?**,**?**] aim to suppress the influence of deformation on discriminative performance, which again does not offer an explicit deformation model. (3) Other work that explicitly takes deformations into consideration [**?**,**?**,**?**] has a narrow focus on individual local tangent spaces, neglecting the relations between them. As we will show, there are significant dependencies between the different tangent spaces of the deformation manifolds, which, if appropriately exploited, contribute greatly to learning a model of deformation.

In this paper, we propose a new approach to deformation modeling, where each observed image is considered to be generated by deforming an object template. The observation of typical deformation patterns exhibited in general images leads to the belief that most deformations are well modeled by a low-dimensional Lie group, which can be characterized by a basis of the associated Lie algebra. Intuitively, the Lie algebraic basis captures the basic deformation patterns, and each deformation in the group is some combination of them. Generally, a different Lie algebra is associated with differing object templates, which, however, are related to each other via the parallel transport property. Specifically, the Lie algebra for one object template is a transported version of the one for others. The fact that parallel transport is covariant with geometric transformation ensures the consistency of this relation.

Consequently, with the Lie algebraic characterization, the problem of learning deformations reduce to the one of estimating the deformation basis for different object templates. Here, we formulate an optimization problem for estimating these bases from a given set of observed images. In this formulation, two levels of relations are exploited: (1) Observed images are closed to the deformation orbits, *i.e.* the manifold is comprised of all deformed versions of the templates. (2) The basis associated with different templates are constrained by the parallel transport relations. The use of the first relation, which explicitly incorporates deformation into the generative process of an image, clearly sets this work apart from the large amount of prior work (*e.g.* those on image manifold learning) that directly model the image space. Additionally, the use of the parallel transport relation further distinguishes the proposed approach from the methods which focus on local neighborhoods only.

The remainder of this paper is organized as follows. Section **??** reviews existing theoretrical results on deformations. The emphasis is particularly placed on the Lie algebraic characterization and parallel transport. Section **??** formulates the optimization algorithm for estimating the deformation model from observed images. Emprirical results are presented in section **??**, where we compare the proposed method with related methods on character recogition and synthesis, as well as face reconstruction. Discussion of the method and results is provided in section **??**.

2 Related Work

We first briefly review previous work on deformations, which roughly fall into two categories. The first category of methods focuses on estimating global affine transformation. Frey and Jojic [**?**,**?**,**?**] proposes a mixture model, where the space of affine transforms is discretized, and an indicator is used to choose a specific transform in generating each image. Miller *et. al* [**?**] proposed a nonparametric probabilistic model, which estimates the global affine transforms by gradually aligning the images, using gradient descend.

The second category takes into account non-rigid deformations that can lead to changes in shapes. Cootes *et. al* [**?**,**?**] proposed the active appearance model

for object alignment, where the deformation is represented via the displacement of pre-specified control points. In addition, approaches by directly matching local descriptors are also widely used. Belongie et al. [**?**] developed a direct matching method using local shape context based on statistics of edges. Keyser et al. [**?**] developed an Image Distortion Model (IDM) [**?**], which pursues a dense match of local patches between two images as a representation of the deformation. Though simple, this method leads to substantial improvement on character recognition, manifesting the important role of local deformations in object recognition. The pioneering work by Tenenbaum etal [**?**] and Roweis and Saul [**?**] triggered a vast amount of work that directly models the image manifold, trying to embed it into local dimensional space.

While deformation information is made use of in building object metrics by the work mentioned above, these approaches do not establish a explicit model of deformations. Recently, new models have been proposed to address this issue. Simard et al. [**?**] considered the manifold of deformations, and tried to approximate it via local tangent spaces. In this work, the basis of these tangent spaces are hand-crafted, with some apparent deformation patterns taken into account (*e.g.* rotation and changes of thickness). However, some subtle variations of shapes are difficult to be captured via manually devised patterns. Another drawback of this method lies in the introduction of tangent spaces for all training samples, incurring unnecessary computational costs in both training and testing phases when the samples are dense. Hastie and Simard [**?**,**?**] improve upon this method by grouping nearby samples into clusters and deriving the tangent basis via learning. However, the learning is done independently for each tangent space, utilizing only the samples within a local neighborhood. This makes it difficult to obtain reliable estimations.

3 The Theory of Deformation

Generally, the shape and size of an non-rigid object can change over time. Such a change is often referred to as a *deformation*, which is ubiquitous in vision problems. In this paper, we focus on the two-dimensional image space, where a deformation can be formalized as a *diffeomorphic transform* on the image plane.

3.1 Lie Group and Lie Algebra

Deformations typically observed in vision problems are a subset of all diffeomorphic transforms, which we assume constitute a Lie group of dimension K. A Lie group G is a finite-dimensional manifold with an algebraic group structure, meaning that it has the following properties:

1. The identity transform is in G.
2. If T_1 and T_2 are both in G, then the composition $T_1 \circ T_2$ is also in G.
3. For each transform $T \in G$, the inverse transform T^{-1} also exists in G.

The Lie group G is associated with a Lie algebra $\mathfrak{g}$, a vector space of dimension K. Each vector $V \in \mathfrak{g}$ is a velocity field and corresponds uniquely to a transform $T \in G$ via the exponentiation mapping as below

$$T = \exp(V). \tag{1}$$

Here, V is called the *Lie algebraic representation* of T.

The relations between a Lie group G and its associated Lie algebra $\mathfrak{g}$ can be described through the construction of a continuous transformation process. Let $V \in \mathfrak{g}$, then for every $t > 0$, $T_t = \exp(tV)$ is a transform. Hence, the function below defines a trajectory on the image plane.

$$\mathbf{x}(t) = T_t \mathbf{x}_0 = \exp(tV)\mathbf{x}_0. \tag{2}$$

Intuitively, this trajectory can be generated through a continuous transformation process described as follows. Consider a particle starting from $\mathbf{x}_0$. If the particle travels across the image plane, passing through each location $\mathbf{x}(t)$ with velocity $V(\mathbf{x}(t))$, the resultant trajectory is then given by

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_{\tau=0}^{t} V(\mathbf{x}(\tau))d\tau. \tag{3}$$

This provides a detailed characterization of the trajectory defined in Eq.(**??**), namely $\exp(tV)\mathbf{x}_0$. Therefore, the transform $\exp(tV)$ can be understood as an operation that sends each point to move for time t, following the velocity field V. Equivalently, the trajectory is characterized by the differential equation below

$$\frac{d\mathbf{x}(t)}{dt} = V(\mathbf{x}(t)). \tag{4}$$

Given a basis of $\mathfrak{g}$, denoted by $\mathcal{B} = (B_1, \ldots, B_K)$, each Lie algebraic vector $V \in \mathfrak{g}$ can be expressed as a linear combination as $V = \sum_{k=1}^{K} \alpha^k B_k$. Intuitively, each base vector of $\mathfrak{g}$ reflects a basic deformation pattern, and all deformations in G are combinations of such base patterns. The Lie algebraic characterization provides a representation, where such combinations can be done via linear operations, great simplifying the modeling and estimation.

3.2 The Action on Images

A deformation $T \in G$ can act on an image by moving the locations of pixels. Let I be an image. Applying T to I results in an deformed image $T \circ I$, given by

$$(T \circ I)(\mathbf{x}) = I(T^{-1}\mathbf{x}). \tag{5}$$

This means that the pixel value of $T \circ I$ at $\mathbf{x}$ equals that of I at $T^{-1}\mathbf{x}$. Let $V \in \mathfrak{g}$. Applying a continuous transform process $\exp(tV)$ to the image I yields a continuous sequence of images, as

$$I_t(\mathbf{x}) = (\exp(tV) \circ I)(\mathbf{x}) = I(\exp(-tV)\mathbf{x}). \tag{6}$$

Taking the derivative *w.r.t.* t, we get

$$\left.\frac{dI_t(\mathbf{x})}{dt}\right|_{t=0} = -V(\mathbf{x})^T \nabla I(\mathbf{x}) \triangleq (V \circ I)(\mathbf{x}). \tag{7}$$

Here, $V \circ I$ denotes the *action of* V *on* I, which produces a scalar map, whose value at $\mathbf{x}$ equals the negated inner product between the velocity $V(\mathbf{x})$ and the image gradient $\nabla I(\mathbf{x})$. Clearly, the action of V is a linear operation on I.

Given a basis $\mathcal{B}$, we can write V in form of a linear combination as $V = \sum_{k=1}^{K} \alpha^k B_k$. Consequently, we can rewrite Eq.(**??**) into

$$\left.\frac{dI_t(\mathbf{x})}{dt}\right|_{t=0} = \sum_{k=1}^{K} \alpha^k (B_k \circ I)(\mathbf{x}). \tag{8}$$

This equation establishes the linear isomorphism between the Lie algebraic representation and the image changes due to deformation. In other words, the infinitesimal changes due to a deformation, whose Lie algebraic representation is a linear combination of some base deformations, can be expressed as the same linear combination of the "base changes", *i.e.* those generated by the base deformations. As we would see in next section, we rely on such decomposition for model estimation from given images.

3.3 Parallel Transport

In general, a deformation group is associated with a specific object, which can not be directly applied to a different object (*e.g.* a transformed version of the object). However, one can adapt a deformation group via the *parallel transport* of the associated Lie algebra, enabling its application to different objects.

Consider an object being deformed, which are observed from two different views. The point at $\mathbf{x}$ from the first view is transformed to $\mathbf{x}' = T\mathbf{x}$ from the second view. Suppose this point has velocity $\mathbf{v}$ at $t = 0$ from the first view, then *what is the velocity of the corresponding point, i.e.* $T\mathbf{x}$*, from the second view?* The derivation below shows the answer:

$$\mathbf{v}' := \lim_{\delta t \to 0} \frac{T(\mathbf{x} + \mathbf{v}\delta t) - T(\mathbf{x})}{\delta t} = \mathbf{J}_T(\mathbf{x})\mathbf{v}. \tag{9}$$

Here, $\mathbf{J}_T(\mathbf{x})$ is the Jacobian matrix of T at $\mathbf{x}$. Here $\mathbf{v}'$ is called the *parallel transport* of $\mathbf{v}$ *w.r.t.* the transform T. The parallel transport can be applied to an entire velocity field V, resulting in a new velocity field $T \bullet V$, given by

$$(T \bullet V)(T\mathbf{x}) = \mathbf{J}_T(\mathbf{x})V(\mathbf{x}). \tag{10}$$

The parallel transports are *covariant* with the inducing transforms, meaning that they satisfy two properties below: (1) the parallel transport induced by an identity transform in itself is an identity, and (2) the parallel transport induced by a composition of two transforms equals the composition of the transports respectively induced, as $(T_2 T_1) \bullet V = T_2 \bullet (T_1 \bullet V)$.

4 Model Estimation Algorithm

In this section, we formulate an optimization problem to estimate the deformation groups for a specific class of objects, given a set of images, and thereon derive an algorithm that jointly solves the basis of the deformation groups and the Lie algebraic coefficients for the training samples.

4.1 Two-Level Formulation

Given a set of n images, we first group them into m clusters, using K-medoid, where each cluster has a *center image*. The number of clusters m is choosen via cross validation, such that all samples within a cluster are close enough to the corresponding center. Suppose the i-th cluster contains n_i samples. For this cluster, we use $I_{i,0}$ to denote the center image of this, and $I_{i,j}$ (with $j = 1, \ldots, n_i$) to the j-th non-center image. Here, we consider each center image as the representation of the canonical shape of an object, and other images in the same cluster as generated by deforming the center image.

As discussed in previous section, a deformation group can be characterized by a Lie algebra. Therefore, the problem of learning the deformation groups thus reduces to the one of estimating the Lie algebraic basis for each cluster. Here, we denote the basis for the i-th cluster by $\mathcal{B}_i = (B_{i,1}, \ldots, B_{i,K})$. To estimate these basis, we formulate an optimization problem, of which the objective function comprises two levels of terms.

Within-cluster Level. Applying the deformation group characterized by the Lie algebraic basis $\mathcal{B}_i$ to the image $I_{i,0}$ yields a K-dimensional manifold comprised of all the deformed images, denoted by $G(\mathcal{B}_i) \circ I_{i,0}$, as

$$G(\mathcal{B}_i) \circ I_{i,0} \triangleq \{\exp(V) \circ I_{i,0} : V \in \mathfrak{g}(\mathcal{B}_i)\}. \tag{11}$$

Here, $\mathfrak{g}(\mathcal{B}_i)$ denotes the Lie algebraic space spanned by $\mathcal{B}_i$. With the assumption that $I_{i,j}$ is generated by deforming $I_{i,0}$, we expect that the $I_{i,j}$ is close to $G(\mathcal{B}_i) \circ I_{i,0}$. Particularly, the distance from $I_{i,j}$ to $G(\mathcal{B}_i) \circ I_{i,0}$ is given by

$$\operatorname{dist}(I_{i,j}, G(\mathcal{B}_i) \circ I_{i,0}) = \min_{\boldsymbol{\alpha}} \left\| I_{i,j} - \exp\left(\sum_{k=1}^{K} \alpha^k B_{i,k}\right) \circ I_{i,0} \right\|. \tag{12}$$

When the deformed image $I_{i,j}$ is close to the center $I_{i,0}$, the coefficients are small. Consequently, by Eq.(**??**), we can approximately write

$$\exp\left(\sum_{k=1}^{K} \alpha^k B_{i,k}\right) \circ I_{i,0} \simeq I_{i,0} + \sum_{k=1}^{K} \alpha^k (B_{i,k} \circ I_{i,0}). \tag{13}$$

As a result, we have

$$\begin{aligned}\mathrm{dist}(I_{i,j}, G(\mathcal{B}_i)\circ I_{i,0})^2 &\simeq \min_{\boldsymbol{\alpha}} \left\| (I_{i,j}-I_{i,0}) - \sum_{k=1}^{K}\alpha^k (B_{i,k}\circ I_{i,0}) \right\|^2 \\ &= \min_{\boldsymbol{\alpha}} \sum_{\mathbf{x}\in\mathcal{D}} \left((I_{i,j}(\mathbf{x}) - I_{i,0}(\mathbf{x})) + \sum_{k=1}^{K}\alpha^k B_{i,k}(\mathbf{x})^T \nabla I_{i,0}(\mathbf{x}) \right)^2 . \end{aligned} \tag{14}$$

Here, $\mathcal{D}$ is the set of all observable pixel locations. For convenience, we define

$$Q_{ij}(\mathcal{B}_i, \boldsymbol{\alpha}_{i,j}) = \left\| (I_{i,j}-I_{i,0}) - \sum_{k=1}^{K}\alpha^k_{i,j} (B_{i,k}\circ I_{i,0}) \right\|^2 . \tag{15}$$

Note that Q_{ij} is a quadratic *w.r.t.* $\boldsymbol{\alpha}_{i,j}$. Hence, the optimal coefficients that yield the minimum (approximate) distance can be readily solved, given $\mathcal{B}_i$.

Inter-Cluster Level. The basis associated with different groups are related to each other via parallel transport. Specifically, we establish a higher-level network between cluster centers, where each center image is connected to several *neighboring centers*, *i.e.* other centers that are not too far from it, such that the optical flow between them can be reliably estimated.

For each pair of neighboring centers $I_{i,0}$ and $I_{i',0}$, we estimate the dense correspondence between them $T_{ii'}$ and $T_{i'i}$, using an optical flow algorithm [need a citation]. Ideally, we would expect the basis $\mathcal{B}_i$ to be the transported version of $\mathcal{B}_{i'}$ *w.r.t.* the transform $T_{i'i} = T_{ii'}^{-1}$, *i.e.* $B_{i,k} = T_{ii'}^{-1} \bullet B_{i',k}$, and vice versa. As some errors may be incurred in optical flow estimation, we use the quadratic term as follows to penalize the deviation from this relation:

$$H_{ii'}(\mathcal{B}_i, \mathcal{B}_{i'}) = \sum_{k=1}^{K} \| B_{ik} - T_{ii'}^{-1} \bullet B_{i',k} \|^2 \tag{16}$$

Here, we have

$$\| B_{ik} - T_{ii'}^{-1} \bullet B_{i,k} \|^2 = \sum_{\mathbf{x}\in\mathcal{D}\cap T_{ii'}(\mathcal{D})} \| B_{ik}(\mathbf{x}) - \mathbf{J}_{T_{ii'}}(T_{ii'}(\mathbf{x})) B_{i'k}(T_{ii'}(\mathbf{x})) \|. \tag{17}$$

Here, $T_{ii'}(\mathbf{x})$ is the location of the pixel on $I_{i',0}$ that corresponds to the pixel at $\mathbf{x}$ of $I_{i,0}$. In general, $T_{ii'}(\mathbf{x})$ does not yield integer coordinates. Under such circumstances, linear interpolation can be used to derive the values of $\mathbf{J}_{T_{ii'}}(T_{ii'}(\mathbf{x}))$ and $B_{ik}(T_{ii'}(\mathbf{x}))$. In addition, $\mathbf{x} \notin T_{ii'}(\mathcal{D})$ indicates that the pixel at $\mathbf{x}$ of $I_{i,0}$ is transformed outside of the observable region, and thus the corresponding term is not included.

Joint Formulation. Integrating the terms at both levels, we derive the joint objective function as follows.

$$L(\mathcal{B}) = \sum_{i=1}^{m}\sum_{j=1}^{n_i} \min_{\boldsymbol{\alpha}} Q_{ij}(\mathcal{B}_i, \boldsymbol{\alpha}) + \gamma \sum_{i=1}^{m}\sum_{i' \in \mathcal{N}_i} H_{ii'}(\mathcal{B}_i, \mathcal{B}'_i). \tag{18}$$

Here, $\mathcal{N}_i$ is a set consisting of the indices of $I_{i,0}$'s neighboring centers, and γ is a positive weight that controls the contribution of the parallel transport constraints. To minimize this function, we introduce an auxiliary function that involves $\boldsymbol{\alpha}_{i,j}$ as arguments:

$$L_{aux}(\mathcal{B}, \boldsymbol{\alpha}) = \sum_{i=1}^{m}\sum_{j=1}^{n_i} Q_{ij}(\mathcal{B}_i, \boldsymbol{\alpha}_{i,j}) + \gamma \sum_{i=1}^{m}\sum_{i' \in \mathcal{N}_i} H_{ii'}(\mathcal{B}_i, \mathcal{B}'_i). \tag{19}$$

Obviously, L_{aux} gives an upper bound of L and has

$$L(\mathcal{B}) = \min_{\alpha} L_{aux}(\mathcal{B}, \boldsymbol{\alpha}). \tag{20}$$

Consequently, $L(\mathcal{B})$ can be optimized by alternating the updates of $\boldsymbol{\alpha}$ and $\mathcal{B}$:

$$\hat{\boldsymbol{\alpha}}_{i,j}^{(t)} \leftarrow \operatorname*{argmin}_{\boldsymbol{\alpha}} \; Q_{ij}(\mathcal{B}_i^{(t-1)}, \boldsymbol{\alpha}), \tag{21}$$

$$\hat{\mathcal{B}}_i^{(t)} \leftarrow \operatorname*{argmin}_{\mathcal{B}} \; \sum_{j=1}^{n_i} Q_{ij}(\mathcal{B}, \boldsymbol{\alpha}_{i,j}^{(t)}) + \gamma \sum_{i' \in \mathcal{N}_i} H_{ii'}(\mathcal{B}_i, \mathcal{B}_{i'}). \tag{22}$$

Note that the value of $L_{aux}(\mathcal{B}, \boldsymbol{\alpha})$ decreases with each updating step. Particularly, the values of L and L_{aux} become equal each time when $\boldsymbol{\alpha}$ is updated to the optima, *i.e.* $L(\mathcal{B}^{(t)}) = L_{aux}(\mathcal{B}^{(t)}, \boldsymbol{\alpha}^{(t+1)})$.

4.2 Initialization

While L_{aux} is convex *w.r.t.* $\mathcal{B}$ and $\boldsymbol{\alpha}$ respectively, this is not a convex optimization problem jointly. Hence, appropriate initialization is crucial as to obtaining a reasonably good solution. Here, we describe a simple yet effective scheme to initialize the the basis $\mathcal{B}$.

We choose a particular cluster as the "standard cluster", and compute the optical flow from the center of this standard cluster to the centers of other clusters. With these optical flows, we can warp the images in other clusters towards the standard one. For example, suppose $I_{1,0}$ is selected as the standard, and the optical flow from $I_{1,0}$ to $I_{2,0}$ is T_{12}, then we warp each image in the second cluster as $I'_{2,j} = T_{12}^{-1}(I_{2,j})$ for $j = 0, 1, \ldots, n_2$. In this way, for each non-standard cluster, we acquire a warped center as well as a set of warped images, which are considered as generated by deforming the warped center.

At the initialization stage, we assume that the standard cluster and the warped clusters share the same basis. To estimate this basis, we compute the

optical flow fields from the standard center to other images in the standard cluster, and those from each warped center to other images in the corresponding cluster. All these flow fields can be roughly considered as residing near the space spanned by the shared basis. Therefore, the basis can be estimated by applying principal component analysis (PCA) to these optical flow fields pooled together. After the basis associated with the standard cluster have been initialized as above, the basis for other clusters can be readily obtained via parallel transport.

5 Experiments

In this section, we discuss results obtained using the previously described deformation model in action space. We first show the generalization power of our model on handwritten recognition task with small amount of training examples, which is similar to the idea of one shot learning [**?**]. Later we test the generative side of our model by sampling new images of digits from our trained deformation manifold and reconstructing a human face given observations of the face from different views.

5.1 Handwritten Digit Recognition

On the popular MNIST [**?**] dataset, several state-of-the-art algorithms perform with an approximate error rate of 0.5%, closed to that of human. Our focus here is two fold. The first is to analyze how current best algorithms deal with deformation modeling and the second is to show that with a more structured deformation model over action space, we can have more generalizablity with a small number of prototypes.

So far, the three most successful algorithms are Support Vector Machine (SVM), Convolutional Neural Nets (CNN) and K-Nearest Neighbor (KNN) with well engineered features and distance functions. The former two algorithms, SVM [**?**] and CNN [**?**], try to form highly nonlinear boundary between different classes in the appearance model by high order polynomial kernels or by broad and deep nets. In order to capture the rich local deformations of handwritten digits, both algorithms have to add many synthetic images after certain deformation which greatly prolong the training process.

On the other hand, current best performed KNN classifiers: Shape Context (SC) [**?**],Image Distortion Model (IDM) [**?**], Tagent Distance (TD), try to search over synthesized images created along deformation field during test instead of remembering them from training. Effectively, these algorithms effectively build local manifold component around each provided training image. SC estimates the deformation field based on shape context descriptor; IDM reconstructs new images by locally moving patches of the image around. However, the flow field estimated for one prototype during training cannot be directly applied to other prototypes. Thus the deformation manifold learnt are made of independent local manifold patches without connection.

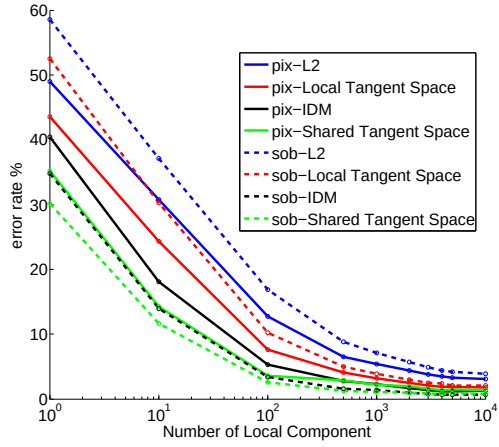

Fig. 1. Filled Line: Recognition error on MNIST dataset with pixel intensity as the feature; Dashed Line: Recognition error on MNIST dataset with response from Sobel filter as the feature.

Though these KNN algorithms make no effort in sharing deformation basis or stiching local manifold structure together, they remain highly competitive on the list. This suggests that test samples in MNIST are mostly covered by the training samples and thus there is little need for finer construction of the deformation manifold. However, if we are limited to a small number of training samples, the sparsely spread information can be gathered by propragating actions learned from one part of the manifold through the constructed network of prototypes.

Below, without any preprocessing the data for better error rate, we compare our algorithms with TD and IDM which are reproduced within certain range of accuraccy. We also add a plain 1-NN classifier with Euclidean distance metric as a baseline measure. We test all these algorithms with varying number of local manifold components to use for classification. For our model, it means different number of prototypes in the first-level network while it is equivalent to different number of training data for the rest algorithms. After reconstructing the test image from deformation manifold, we tested classification error rate based on projection distance using intensity and sobel feature. Figure **??** shows that with small amounts of trarining data, the recognition error rate drops faster than others by sharing information through the connected network.

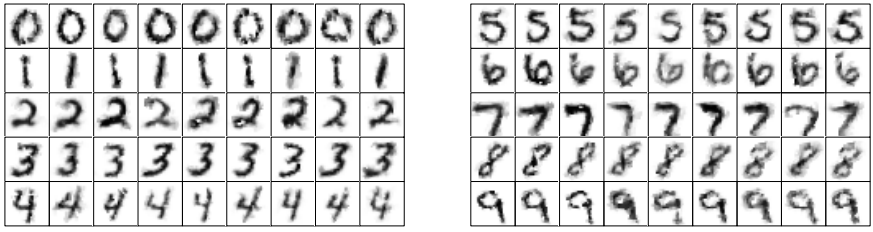

Fig. 2. Synthesized digits from the learned digit deformation manifold. The first digit in the row is the prototype; the rest are locally deformed from the prototype with a random coefficients of the learned basis

5.2 Image Synthesis

The main point of image synthesis is not about super resolution reconstruction appealing to human, but to show that the basis learned shared across the manifold do make sense :p. **Digit Synthesis.** Starting from the digit manifold learned from MNIST dataset, we are now able to synthesize new images of digits by applying randomly sampled action coefficient to randomly sampled prototypes. In Figure **??**, we show the randomly sampled prototype of each digit in the first column and synthesized new digit images in the rest of each row. We can see that the synthesized images generated by integrating along the geodesic of the manifold cannot be explained by global affine deformation. In the row of digit two, we can see that there are some basis related to the size of the lower left circle of the digit.

Face Reconstruction. In addition, we learn a face manifold using Frey Bredan's face dataset, containing around 2,000 20 by 28 gray scale images of Frey's face in different expressions and angles of view. For this experiment, we first learn the commonly shared basis to contruct the manifold from 1,000 sampled images. Then for each of another randomly sampled 500 testing images, we try to see how close it can be projected onto the manifold. We tested three different algorithms for reconstruction: nearest training image in Euclidean metric, closest projection onto tagent spaces and closest projection onto our connected deformation manifold with shared basis. Again, we test our results with varying number of prototypes to use. Shown in Figure **??**, we can see that the in terms of both Euclidean distance and PSNR ratio, the reconstruction from manifold with learned shared basis is consistently better than those learned independently from training examples. Since the images are by themselves small and most reconstruction errors are not human detectable, we do not show the reconstructed faces.

6 Conclusion

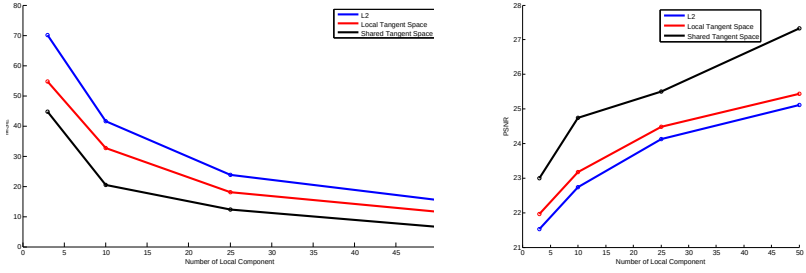

Fig. 3. Synthesized digits from the learned digit deformation manifold