Separating pose and expression in face images: a manifold learning approach

Jihun Hamm and Daniel D. Lee
Department of Electrical and Systems Engineering, University of Pennsylvania
Moore Bldg, 200 South 33rd St, Philadelphia, PA 19104, USA
jhham,ddlee@seas.upenn.edu

Abstract

Digital images of a person’s face display a wide range of variations from differing pose, expression, and illumination conditions. Such variations can be modeled empirically by an appearance manifold in the image space. In this paper, we tackle the problem of learning the appearance manifold of faces in an unsupervised way. In particular, we aim to extract the substructure of facial expressions and the substructure of pose change separately. Two different distances, orbit-distance and group-distance, are defined which measure the difference of images due to expression and pose respectively. To reconstruct the complete structure of the manifold, we collect the local distances and compute a factorized isometric embedding of the original data. We test the proposed method on a known face database and demonstrate its capability of separating expression and pose via unsupervised learning.

1 Introduction

It is known that the set of face images under continuously varying expression, pose, and lighting condition, forms an appearance manifold in the image space $\mathbb{R}^D$ [4, 15, 14, 18]. A compact representation of such appearance manifold has many potential uses, including human–computer interaction, facial animation, and the recognition of multiple subjects. In this paper we are interested in finding a low-dimensional representation of the manifold where pose and expression are explicitly factorized.

In general the appearance manifold does not lie on a linear subspace, and cannot be modeled analytically. A promising approach to the modeling problem is to learn the nonlinear structure of the appearance manifold empirically. Manifold learning techniques can learn the nonlinear geometry or probabilistic density of data samples, and can interpolate the data smoothly to generate novel data. Much progress has been made in manifold learning field since the publication of novel dimensionality reduction algorithms such as Isomap [20, 6], Locally Linear Embedding (LLE) [16, 17] and Laplacian Eigenmap [3]. However, direct applications of dimensionality reduction algorithms on face images do not achieve the desired separation: 1) typically the resultant embedding is confounded by various sources and requires human intervention for interpretation and, 2) the algorithms can totally fail to discover the subtle structure of expression, buried by the more conspicuous structure due to pose and illumination change. We will demonstrate the case in a later section using the image database shown in Fig. 1.

A simple attempt to reduce the confounding effect is to register all images to a template image. We can choose a template image among the data, say a frontal-view face with neutral expression, and register all the other images to the template using affine or perspective transforms. By the single-template registration, variations such as in-plane rotation, translation and scaling are mostly suppressed, and the registered images will highlight the expression difference better than the original images. However, the affine/perspective transform can only approximate the rigid motions of face for a limited range of pose close to the neutral pose. To handle a wider range of pose, the registration needs to be applied to all pairs of images, where each image serves as a template. By using multiscale registration techniques...
Figure 1: Images showing five classes of facial expressions from a subject with arbitrary pose variations. Although a human can efficiently distinguish the expressions regardless of the pose, the difference in the pixel intensities of any two picture is dominated by their pose difference rather than their expression difference.

[21], pairwise registration can be performed efficiently. With the registration procedure, we show how to use pairwise registration results to get global information on the appearance manifold, and find a low-dimensional representation of it.

The registration procedure admits a geometric interpretation of invariance: the distance between two images up to transformation, referred to as ‘manifold distance’ or ‘tangent distance’, is the distance of features which are invariant to the transformation. The concept has been applied to pattern recognition [19] and clustering [7, 12], where the usual distance is replaced by the manifold distance or the tangent distance. In [12], a Lie group interpretation of the invariance has been introduced, where the manifold distance is redefined as the orbit distance under a quotient map. The invariant distance was also used for dimensionality reduction [8], which extents the LLE algorithm to find a transformation-invariant embedding of images. However, these approaches only account for invariant structure and ignore the variant structure of the data.

In this paper we take a step further and consider both the variant and the invariant substructures of the data. We adopt the Lie group framework similar to [12] and define another distance, the group distance, to describe the variant substructure. These local measures are combined into intrinsic distances, similarly to Isomap algorithm, from which we find a factorized, low-dimensional embedding of the face images, where pose and expression are explicitly separated.

The remainder of the paper is organized as follows. Sec. 2 introduces isometric embedding methods for learning appearance manifolds. Sec. 3 explains our multiscale registration technique, and the effects of registrations on the geometry of appearance manifolds. Sec. 4 describes the proposed method of learning a factorized embedding and demonstrates the method with real face data. Finally, future work and extensions are discussed in Sec. 5.

2 Learning appearance manifolds by isometric embedding

Images reside in a very high-dimensional space $\mathbb{R}^D$, typically $D = 10^3 \sim 10^4$. Manifold learning algorithms can find a low-dimenional representation from the high-dimensional data, which makes it easy to analyze or visualize the underlying structure of data. In this section we describe the Isomap algorithm and discuss difficulties involved in learning the structure of face images with pose variations.

2.1 Multidimensional scaling and Isomap

Let $X = \{x_1, \ldots, x_N\} \subset \mathbb{R}^D$ be the given data, where $x_i$ is the vector of gray-scale pixel values of $i$-th image, and let $Y = \{y_1, \ldots, y_N\} \subset \mathbb{R}^d$ be the corresponding low-dimensional ($d \ll D$) features through a dimensionality-reducing map $f : X \rightarrow Y$, $y_i = f(x_i)$. If the map $f$ preserves topology then $f$ is called an embedding. We will also use the term embedding to denote the resultant features $\{y_i\}$, following the convention in manifold learning literature.
A well-known method of finding the low-dimensional features is Multidimensional Scaling (MDS) [5], whose goal is find $Y$ which preserves distances $d(y_i, y_j) \approx d(x_i, x_j), \forall i, j = 1, 2, ..., N$. One can define the error between $d(y_i, y_j)$ and $d(x_i, x_j)$ in multiple ways and derive different optimization problems. The simplest implementation of MDS is Classical MDS which uses a spectral decomposition of the distance matrix to compute $y_i$.

The original version of the classical MDS uses Euclidean distances for both $X$ and $Y$. However, if we assume $X$ lies on a low-dimensional submanifold $M \subset \mathbb{R}^D$, then the Euclidean distance of $\mathbb{R}^D$ is not the true distance on the submanifold $M$. Since we are interested in the intrinsic structure of the data and not how the data is embedded in $\mathbb{R}^D$, we require the intrinsic distance, not the Euclidean distance, to be preserved through an embedding $f$. Such an embedding is called an isometric embedding. The intrinsic distance $g$ on a submanifold $M \subset \mathbb{R}^D$ is defined as the length of the shortest path contained in $M$: $g(x_i, x_j) = \inf_{\gamma} \{ L(\gamma) \mid \gamma : [a, b] \to M, \gamma(a) = x_i, \gamma(b) = x_j \}$, where $L(\gamma) = \int_a^b |\gamma'(t)| \, dt$ is the length of the curve $\gamma : [a, b] \to M$. Finding the minimizing curve involves solving differential equations in general. Consequently, even if the manifold is an analytical one, it is nontrivial to solve for such curves exactly.

The major contribution of Isomap was to approximate the shortest-path computation, by 1) finding a K-nearest-neighbor graph and 2) use the shortest-path on the graph instead, which can be done in polynomial time. The algorithm is illustrated in Fig. 2 and summarized below.

**Classical MDS:**

1. Compute the inner product matrix $[G_{ij}]$ by $G_{ij} = \frac{1}{2} (d^2(x_i, \bar{x}) + d^2(x_j, \bar{x}) - d_{ij}^2)$, where $\bar{x}$ is the center of $X$, and $[d_{ij}]$ is the given distance matrix.

2. Compute the $d$-largest eigenvectors $Z = [z^1, ..., z^d]$ of $[G_{ij}]$.

3. Embedding $y_i$ is given by $i$-th row of $Z$ as $y_i = [z^1_i, ..., z^d_i], \forall i = 1, ..., N$.

**Isomap:**

1. Construct a K-nearest neighbor graph from the given distance matrix $[d_{ij}]$.

2. Compute the lengths $g_{ij}$ of the shortest-paths on the graph by dynamic programming [20].

3. Apply classical MDS on $[g_{ij}]$. 

\[\begin{array}{c}
\text{Figure 2: Left figure shows a plot of 1000 sample points on a horseshoe-like surface. For any two points (dark dots) on the submanifold, the Euclidean distance of the direct path (blue dashed line) in the ambient space is different from the intrinsic distance of the geodesic curve (blue real curve) on the submanifold. In the middle figure, sample points are plotted with the K-nearest neighbor graph (K=8.) Isomap algorithm approximates the geodesic curve (blue real line on the left figure) with the shortest path on the K-nearest neighbor graph (red real curve on the middle figure.) On the right is the final two-dimensional embedding recovered by the algorithm. The shortest path (red real curve) is a good approximation of the geodesic path (blue real curve). These figures are analogous with Fig. 3 in [20] with a different data set.}
\end{array}\]
2.2 Application to face images

We first explain the database used throughout the paper. The images used in our experiments are from the Cohn-Kanade face database which consists of sequences of six different prototypic facial expressions (joy, surprise, anger, fear, disgust, and sadness) for over 200 subjects [10]. We have chosen five expressions of a single subject in the database. Each expression consists of an eight-image sequence varying from slight- to full-expression. Since the original data does not have annotated pose variations, we have simulated ten poses for each image in the sequence. This was done by a random rigid movement of the head (yaw angle=±18°, pitch angle=±14°, roll angle=±7°) under an ellipsoidal head model. After each image is cropped to 50 by 50 pixels with background removed, we have a total of $N=400$ images of $D=2500$ dimensions with arbitrary pose and expression. A subset of the data are shown in Fig. 1.

The Isomap algorithm has been demonstrated on various types of images in the literature. However, when it is applied to our data, the algorithm cannot find the structure of expression, as shown in Fig. 3. The neighborhood size is $K=12$, and similar results are observed for different neighborhood sizes. A further analysis of the result reveals the crux of the matter: the image difference due to expression is much smaller than the difference due to pose. Consequently, for a small $K$, the K-nearest-neighbor consists exclusively of the images of the similar pose regardless of the expression, and conversely two images of the same expression but of different poses are not in the same K-nearest-neighborhood, preventing the algorithm to find the local structures of expression. On the other hand, as $K$ increases, the algorithm simply behaves like a linear MDS and loses its ability to learn a nonlinear structure.

To remedy the situation, we need to understand how pose change affects the geometry of data and to incorporate the knowledge into the algorithm. We will first review in the next section how image registration can help the algorithm to achieve invariance to pose change. A full-fledged model will be proposed in Sec. 4.

3 Image registration and invariance

The change of appearance of a face due to pose change can be approximated by warping the image with affine/perspective transforms. In this section we explain our multiscale image registration
Figure 4: Given a target and a source images, multiscale registration finds the best registration parameters in the coarsest resolution (level 3), and uses the parameters to initialize the registration on the next resolution (level 2), and repeats the procedure until it reaches the finest resolution (level 1).

technique, and introduce the use of the registration to achieve invariance.

3.1 Multiscale image registration

Let \( x(u, v) \) be an image, and \( T \) be an affine/perspective transform of 2D coordinates \((u', v') = T(u, v)\). Then the warping \( T \) on \( x \) is defined as \( y(u, v) = x(T^{-1}(u, v)) \). Image registration is the procedure of finding the best warping of an image that matches another image. We use the well-known gradient-based registration technique [13] to avoid the problems involved in feature detection and correspondence matching.

Given two images \( x \) and \( y \), the registration of \( y \) to \( x \) is the minimization problem of the sum of squared differences

\[
\min_T \sum_{u,v} |x(u,v) - y(T^{-1}(u,v))|^2.
\]  

(1)

Since the squared error usually has multiple local minima, the optimization is sensitive to initialization of parameters. To relieve the problem, we use multiscale image registration [21]. In multiscale registration, image pyramids of multiple resolutions are generated from both the source and the target images, and registration is performed from the coarsest to the finest levels of the pyramids. At each level, registration is actually done by iterating the Gauss-Newton method described below. Let \( T_g \) denote the transform parameterized by \( g \in \mathbb{R}^k \), and let \( C(x,y) \) be the squared distance

\[
C(x,y) = \sum_{u,v} |x(u,v) - y(T_g^{-1}(u,v))|^2.
\]  

(2)

The Gauss-Newton method finds the minimum of Eq. 2 by the update rule:

\[
g^{n+1} \leftarrow g^n - \alpha (\nabla^2_g C|_{g^n})^{-1} \nabla_g C|_{g^n},
\]  

(3)

where \( \nabla_g C \) and \( \nabla^2_g C \) are the gradient and the Hessian of \( C \), respectively. These are computed from the first and the second order derivatives of \( T \) with respect to \( g \), and the first and the second order derivatives of images with respect to coordinates \((u,v)\). The coefficient \( \alpha \) determines the speed of update \((0 < \alpha \leq 1)\). Fig. 4 shows an example of multiscale registrations applied to a pair of face images.

**Multiscale image registration:**

1. Given a source image \( y \) and a target image \( x \), derive image pyramids for both.
2. Choose initial value \( g^0 \), and start from the coarsest level.
3. Update \( g^{n+1} \) by Eq. 3.
4. Repeat 3 until convergence.
5. Go to one finer level and repeat 3 - 4.
Figure 5: An appearance manifold $M$ is assumed to decompose into the invariant substructure $M/G$ and the variant substructure $G$ under a principal bundle model $M = M/G \times G$. Those substructures are learned from the data by measuring the orbit distance and the group distance defined as follows: if $x$ and $y$ are two images on the manifold $M$ with different poses and expressions, then, the orbit distance $d_{\text{orb}}$ measures the difference in expression, and the group distance $d_{\text{grp}}(x, y)$ measures the difference in pose.

### 3.2 Group action and invariance

A straightforward use of image registration in conjunction with isometric embedding is to register all images to a single template, and apply the algorithm on the registered images. As noted before, single-template registration can remove most of the in-plane rotation, translation and scaling variations in the data. However, this procedure is often not enough to account for all pose changes, and the remaining error still hinders the algorithm from discovering the expression structure correctly.

Since single-template registration has limited capability, we use multiple templates for registration. As an extreme case, we consider registering all images to all other images in the database, that is, the registration is performed for all pairs. The information we get from the pairwise registration is redundant, leaving the problem of how to use the pairwise registration results to get the expression information which is invariant to pose change. To find a solution, we need a better understanding of how transformations affect the geometry of appearance manifold.

A geometric framework for invariance has been proposed in [12], through the use of group actions and quotient spaces. If a transformation group $G$ acts on a manifold $M$, then one can associate the action with an abstract topological space $M/G$, the quotient space. The space $M/G$ conceptualizes the invariant structure of the data, in that a quantity defined on $M$ that is invariant to the action of $G$, can be naturally defined as a derived quantity in the quotient space $M/G$. In [12], the orbit distance, which is an invariant distance of two images under affine transformation, was defined and used for affine-invariant clustering of images. Computationally, the orbit distance between two images, which we will detail in the next section, is computed from the residual distance between the two after registering one image to another. In fact, more can be said about the differential geometry of appearance manifold under Lie group action. The appearance manifold can be assumed to be factorized under a Lie group action, into the variant and the invariant substructures, corresponding to pose and expression variability, respectively. By a novel use of an isometric embedding algorithm, we can reconstruct the factorized structure of face images from the data. We will present the theoretical and algorithmic framework in the next section.

### 4 Factorized isometric embedding

In Sec. 2 and Sec. 3, we introduced the basics of isometric embedding algorithm and group theoretic understanding of invariance. In this section we define the orbit and the group distances to completely describe the structure of appearance manifold under a Lie group action, and use isometric embedding to achieve a factorized embedding of the data in terms of pose and expression.
4.1 Group and orbit distances

The set of images resulting from all $G$-actions on an image $x$, that is $\{g \cdot x | g \in G\}$, is called the orbit of $x$ and is denoted by $G \cdot x$. For example, if $x \in M$ is an ‘happy’ face at a particular pose, then the orbit $G \cdot x$ is set of all ‘happy’ faces of arbitrary poses. The geometry of $M$ under the group action can be studied from its orbits structure. Well-known theorems in Lie group theory guarantee that under plausible assumptions each orbit is diffeomorphic to the transformation group $G$, and the quotient structure $M/G$ is another manifold in itself [11, 22]. This implies the appearance manifold $M$ is locally a principal bundle $M \cong M/G \times G$ (details are given in [9].) We further assume a global principal bundle $M = M/G \times G$ as a model of the appearance manifold $M$. The factorized structure $M/G \times G$ can be learned from data by measuring two types of distances, the orbit distance and the group distance. These two distances completely characterize the invariant $M/G$ and the variant $G$ substructures of $M$. The concepts are illustrated in Fig. 5. If $x$ and $y$ are two points on the manifold $M$, then the orbit distance $d_{\text{orb}}$ is defined as

$$d_{\text{orb}}(x, y) = \min_{g \in G} d(x, g \cdot y),$$

where $d(\cdot, \cdot)$ is a Euclidean distance. From the minimizer $g$ above, the group distance $d_{\text{grp}}$ is defined as

$$d_{\text{grp}}(x, y) = \|T_g - T_e\|,$$

where $\|T_g - T_e\|$ is the 2-norm of the displacement field of the transform $T_g$ from the identity transform $T_e$. Computationally speaking, the unique transform $g$ of Eq. 4 is found by registering the image $y$ to the image $x$, and once the optimal warping $g$ is found, the amount of the warping defines the group distance.

Note that $d_{\text{orb}}$ is defined from a Euclidean distance $d(\cdot, \cdot)$ on $M$. As $M$ is a nonlinear manifold in general, the $d(\cdot, \cdot)$ differs from the true intrinsic distance on $M$. Consequently, the $d_{\text{orb}}$ is not the intrinsic distance on $M/G$ either. To compute the intrinsic distance $g_{\text{orb}}$ out of $d_{\text{orb}}$, we resort to the idea of Isomap: a K-nearest neighbor graph is constructed from $d_{\text{orb}}$, and the global orbit distance $g_{\text{orb}}$ is computed from the length of the shortest-path on the K-nearest neighbor graph. Similarly, the global group distance $g_{\text{grp}}$ is computed from the shortest-path on the K-nearest neighbor graph, where $g_{\text{grp}}$, instead of $d_{\text{orb}}$, is used for the length of an edge of the graph.

The two intrinsic distances $g_{\text{orb}}$ and $g_{\text{grp}}$ characterize the invariant $M/G$ and the variant $G$ structure of data. By embedding $g_{\text{orb}}$ and $g_{\text{grp}}$ isometrically to low-dimensional Euclidean spaces, denoted by $E_{\text{orb}}$ and $E_{\text{grp}}$, we can reassemble the principal bundle $M = M/G \times G$ in a Euclidean space by taking the product $M = E_{\text{orb}} \times E_{\text{grp}}$. Since the embeddings $E_{\text{orb}}$ and $E_{\text{grp}}$ represent the expression and the pose structures of the data respectively, the Cartesian product $E_{\text{orb}} \times E_{\text{grp}}$ is the desired factorization of data into expression and pose. The algorithm is summarized below.

**Factorized isometric embedding:**

1. For $i, j = 1, ..., N$, compute the local distances $d_{\text{orb}}(i, j)$ (and $d_{\text{grp}}(i, j)$) by multiscale registration, using affine or perspective transforms.
2. Form a K-nearest neighbor graphs from $d_{\text{orb}}$ (and $d_{\text{grp}}$.)
3. Compute the shortest-path distances $g_{\text{orb}}$ (and $g_{\text{grp}}$) from $d_{\text{orb}}$ (and $d_{\text{grp}}$), using the K-nn graph.
4. Find isometric embedding $E_{\text{orb}}$ (and $E_{\text{grp}}$) by applying MDS on $g_{\text{orb}}$ (and $g_{\text{grp}}$.)
5. Combine $E_{\text{orb}}$ and $E_{\text{grp}}$ to get the final factorization $E_{\text{orb}} \times E_{\text{grp}}$.

4.2 Experiments

We use the same images from the database explained in Sec. 2.2 to test the factorized embedding algorithm. The size of the neighborhood was $K = 10$.

$E_{\text{orb}}$ represents the invariant structure $M/G$ corresponding to expression change, and we indeed observe the structure in Fig. 6. As a comparison, the naive Isomap embedding, shown in Fig. 3, could not discover the structure of facial expression at all. Similarly, $E_{\text{grp}}$ represents the purely variant structure $G$
corresponding to pose change, which also agrees with the result in Fig. 7. Quantitatively, the normalized correlation between the first coordinates $E_{\text{grp}}^1$ and the true pitch angle was 0.963, and the normalized correlation between the second coordinates $E_{\text{grp}}^2$ and the true yaw angle was 0.973.

Finally, a two-way factorization of the data is achieved by taking the first two coordinates $E_{\text{orb}}^1, E_{\text{orb}}^2$ and the first coordinates $E_{\text{grp}}^1$ and combining them into a three-dimensional data. The combined embedding represents the desired factorization of data by (expression $\times$ pose), and is shown in Fig. 8.

The computation took about 30 minutes on a Pentium 4 machine with 2GB of memory. The current implementation of the algorithm is written in Matlab. We expect the computation time to be significantly reduced with the use of optimized and precompiled binaries in upcoming implementations.

5 Conclusion

Manifold learning algorithms are promising alternatives to conventional learning methods when the structure of data has to be learned empirically from the appearance. These algorithms are especially useful for analyzing high-dimensional data such as image databases. In this paper, we proposed a factorized isometric embedding algorithm which can handle face images with pose and expression variations. By exploiting the knowledge of the geometry of appearance manifolds under Lie group action, the algorithm can separate expression and pose from the data without further information. The potential of the algorithm has been demonstrated on a subset of Cohn-Kanade database, and is currently under evaluation for larger data sets.

One of the remaining challenges in learning the appearance manifold of faces is to handle the illumination variations. For a Lambertian surface, the set of images of all illumination conditions has been shown to lie on a cone or a subspace [2, 1] if the pose remains fixed. However, there has been little work on the effect of the simultaneous change of pose and illumination on the geometry of the appearance manifold. In future work, we aim to model the illumination change as a group action and to incorporate it into the proposed factorization model.

References

Figure 7: Left and middle figures show the same two dimensional embedding $E_{grp}$, which represents the group $G$ of rigid motions. The two figures are color-coded by the true value of yaw and pitch angles respectively. In the right figure, the original images are superimposed on the embedding. One can check that the yaw angle changes along the horizontal axis and the pitch angle changes along the vertical axis.


Figure 8: Left figure shows the three-dimensional embedding $E_{\text{orb}}^1 \times E_{\text{orb}}^2 \times E_{\text{grp}}^1$. On the right, three random slices of the embedding along the third axis are shown, with the original images superimposed. The pitch angle of the face increases as the slice traverses from front to back. However, the faces in each slice are arranged by their expression consistently.

Jihun Hamm received his B.S. and M.S. in electrical engineering from Seoul National University, and is currently a Ph.D. candidate in the Dept. of Electrical and Systems Engineering at the University of Pennsylvania. His research interests include manifold learning algorithms and applications thereof in computer vision and robotics. (Homepage: http://www.seas.upenn.edu/~jhham)

Daniel D. Lee received his A.B. in physics from Harvard in 1990, and his Ph.D. in condensed matter physics from MIT in 1995. He is currently an Associate Professor in the Dept. of Electrical and Systems Engineering at the University of Pennsylvania. His research focuses on applying knowledge about biological information processing systems to building better artificial sensorimotor systems that can adapt and learn from experience. (Homepage: http://www.seas.upenn.edu/~ddlee)