Identity- and Illumination-Robust Head Pose Estimation Using Manifold Learning

Chao Wang, Xubo Song
Department of Biomedical Engineering, Oregon Health and Science University

Abstract - Head pose estimation using manifold learning is challenging due to other appearance variations such as identity and illumination changes. To address the problem, we propose to incorporate supervised information (pose angles of training samples) into the process of manifold learning. Most manifold learning algorithms have two common variables: inter-point distances and graph weights, which can greatly affect the property of the constructed manifold. We propose to redefine these variables by constraining them with the pose angle information. In addition, since the environmental illuminations are distributed in the low-frequency component of the image and the texture-based feature is irrelevant to the pose variation, we use the proposed Localized Edge Orientation Histogram (LEOH) rather than the pixel intensity feature for manifold learning. The experimental results show that our method has the highest estimating accuracy and is robust to identity and illumination.

Keywords: Robust head pose estimation, supervised manifold learning, localized edge orientation histogram, inter-point distance, graph weight

1. Introduction

Head pose estimation [1], the automatic estimation of head orientation related to a camera-centered coordinate system, is very important in many applications, such as driver monitoring, 3D face modeling, face recognition and human-computer interface. Previous algorithms on head posed estimation can be divided into five categories: (1) shape-based geometric analysis [2], where head pose is deduced from geometric information like the configuration of facial landmarks; (2) model-based methods [3], where non-linear parametric models are derived before using a classifier; (3) appearance-based approaches [4], where the pose estimation problem is viewed as a pattern classification problem on image feature spaces; (4) template matching approaches [5], which are largely based on nearest neighbor classification against texture templates; (5) manifold learning and dimensionality reduction [6-11], where the manifold represents the underlying geometry structure of the pose space.

The focus of this paper is head pose estimation based on manifold learning. General manifold learning algorithms, which map the high-dimensional data to a low-dimensional space while preserving the global geometrical structure of the datasets, have four common stages:

1. Feature representation of data point: Each data point (face image) can be represented by a feature vector $\mathbf{x}_i$, which can be defined as, for instance, grayscale pixel intensity or histogram-based feature. This forms the basis for the subsequent stages.

2. Neighborhood construction based on inter-point distance: For each data point $\mathbf{x}_i$, if point $\mathbf{x}_j$ is one of its $K$ nearest neighbors by measuring the inter-point distance based on the feature defined in Stage 1, then connect them to form a neighborhood. Therefore, the inter-point distance is a crucial factor to construct the neighborhood graph.

3. Graph weight computation: For any two connected data points, we set a graph weight, which represents the degree of closeness between the two points. The weights in a neighborhood characterize their local geometry structure.

4. Projection computation: Given the neighborhood graph, a nonlinear or linear projection maps the high-dimensional data to a low-dimensional space. The mapping preserves the intrinsic geometry of the underlying manifold by optimizing an objective function.

This paper improves upon the first three stages of manifold learning. Specially, we use a feature representation that is robust to identity and illumination (Stage 1). We also incorporate supervised pose information into the definition of inter-point distance and graph weight (Stage 2 & 3).

There have been efforts for robust feature representation in Stage 1. Raytchev et al. [6] used a derivative of Gaussian filter to enhance the edge of face image while Balasubramanian and Ye [9] adopted the Laplacian of Gaussian transformed image feature. Both methods can construct an edge map, which is useful since pose variation in face image is a direct result of geometric transformation and not relative to texture information [6]. In addition, the environment illumination is treated as the low-frequency component of the image [12]. On the contrary, the edge-based feature is high-frequency, which is insensitive to illumination changes. However, their methods only used the pixel intensity of the edge map for feature representation, which is not enough to describe the edge changes, such as orientation, among different head poses. We propose a Localized Edge Orientation Histogram (LEOH) based feature, which uses a histogram-based descriptor for the edge orientation of the edge map, especially in the local region.

There have also been efforts to incorporate the supervised information (pose angles of the training samples) into Stage 2.
Fu and Huang [8] constructed the neighborhood graph only using the data points with the same pose angle, which is based on classification. However, head pose estimation is a regression problem. Balasubramanian and Ye [9] presented a biased manifold embedding framework in which the head pose information is used to compute a biased neighborhood of each training sample. However, how to determine the biased degree is not clear. We propose to reset the comparison of the inter-point distances as a two-grade structure: Pose difference is the first and feature (e.g., LEOH feature) difference takes the second place.

There are few papers on how to incorporate the supervised information into Stage 3 since the definitions of the graph weights are different in many manifold learning algorithms. We present two ways (direct and indirect) to incorporate the supervised information into the computation of the graph weight, which can be applied to general manifold learning algorithms.

There have also been efforts [10, 11] to incorporate the supervised information into Stage 4, but which is not considered in this paper.

In this paper, the manifold constructed by our method (LEOH feature representation, inter-point distance and graph weight redefinition) strongly relies on the pose changes rather than the other variations, such as identity, illumination, facial expression and background. To evaluate our method, we apply it on four manifold learning algorithms: Locally Linear Embedding (LLE) [13], Laplacian Eigenmaps (LE) [14], Neighborhood Preserving Embedding (NPE) [15] and Locality Preserving Projections (LPP) [16]. The latter two are linear variants of LLE and LE, respectively. Based on these four manifold learning algorithms, we compare our method with the other related methods on two public face databases: FacePix [17] and MIT-CBCL [18] which contains the illumination variations.

2. Localized edge orientation histogram

We present Localized Edge Orientation Histogram (LEOH) feature for manifold learning. The original pixel intensity feature contains not only the shape information but also the texture information. Preliminary experiments conducted with Gabor filters and Fourier-Mellin transformed images indicated that texture-based features are not ideal for pose related problem [9]. In the real world, environment illumination can produce shadow on the face, which changes the pose manifold geometry and thus adversely impact the result of head pose estimation. Environment illumination can be regarded as the low-frequency component of the image. Therefore, we can use high-frequency, edge related information to construct an illumination-robust feature. In addition, the original pixel feature is not relative to the local change during the pose variation. The proposed LEOH feature contains three improved factors: shape-based, edge related and localized. The LEOH feature is inspired by Histogram of Oriented Gradient (HOG) [19]. The HOG descriptor has been successfully applied in human detection. With gradient computation, the HOG feature is an appearance-based feature which also reflects texture change, but LEOH only focus on shape information. For computing LEOH feature, we adopt Canny descriptor for edge detection. The LEOH feature can be computed as follows (see Figure 1):

![Figure 1. Extraction of LEOH feature](image)

(1) We compute the gradient (mask: [-1, 0, 1] and [-1, 0, 1]T) of a gray-level face image without Gaussian smoothing. By experiments, we found that Gaussian smoothing filter can blur the edge information of the face, which reduce the accuracy of head pose estimation.

(2) We detect the Canny edge. Given the gradient orientation θ (0°~360°) of each pixel, the edge angle is quantized evenly into eight parts. Then we use eight filters to detect the corresponding edges. Given the gradient magnitude G (normalized to [0, 1]) of each pixel, the pixel with large gradient magnitude is more likely to correspond to edge than that with small gradient magnitude. However, only a threshold can not specify the characteristic of the pixel. In the Canny algorithm, we use thresholding with hysteresis (big threshold Tb, small threshold Ts) in the edge detection: Start from the pixel with gradient magnitude > Tb and end to the pixel with gradient magnitude > Ts.

(3) The face image is divided into M × N cells and every cell has B histogram bins of the edge pixels. The histogram bins are increased by the gradient magnitude. The gradient orientation θx,y of pixel (x, y) can be quantized into one of the histogram bins Bx,y as follows:

\[
B_{x,y} = \left\lfloor B \times \frac{\theta_{x,y}}{360+\epsilon} \right\rfloor + 1
\]

where ε is a very small positive constant and the discrete level \(B_{x,y} \in [1, B]\).

(4) Like HOG descriptor, we found that block normalization, such as L1-norm, L2-norm and L2-Hys [19], is also
3. Supervised manifold learning

In this section, we redefine the inter-point distance and graph weight of manifold learning algorithms by incorporating the pose angle information of the training samples. We apply the two pose-constrained variables to four manifold learning algorithms (LLE, LE, NPE and LPP).

3.1. Pose-constrained inter-point distance

We propose to set the comparison of the inter-point distances as a two-grade structure. The first or superior grade is the comparison of the pose angles between two points. The second grade is the comparison of the feature-based (Stage 1) distance metric, such as Euclidean, Mahalanobis and cosine, etc. The original inter-point distance of manifold learning algorithms only focuses on the latter grade.

Balasubramanian and Ye [9] presented the Biased Manifold Embedding (BME) framework to adjust the distance. They multiplied a new term to the original inter-point distance, which is the pose similarity between the two data points, described as follows:

\[
\tilde{d}_{ij} = f([z_i - z_j]) \cdot d_{ij}
\]

where \(d_{ij}\) is the original inter-point distance between point \(\tilde{x}_i\) and \(\tilde{x}_j\); \(\tilde{d}_{ij}\) is the adjusted distance; \(z_i\) and \(z_j\) denote the pose angle information (e.g., yaw pose degree) of \(\tilde{x}_i\) and \(\tilde{x}_j\), respectively; \(f\) is an increasing positive function, which is represented as:

\[
f([z_i - z_j]) = \alpha \cdot \frac{|z_i - z_j|}{\max(|z_i - z_j| - |z_i - z_j| + \varepsilon)}
\]

where \(\alpha\) is a positive constant. \(\varepsilon\) can be an arbitrary small positive constant which prevents the denominator of \(f\) to be zero.

However, how to determine the biased degree (the influence of function \(f\) in \(\tilde{d}_{ij}\)) is not clear in BME. We revise the function \(f\) to be an exponential form. The new adjusted distance \(\tilde{d}_{ij}'\) can be defined as:

\[
\tilde{d}_{ij}' = f([z_i - z_j])^n \cdot d_{ij} \quad (n > 1)
\]

where the parameter \(n\) is proportional to the influence capability of \(f\).

In order to construct a pose related manifold, we should incorporate the supervised information (pose angles of the training samples) into the manifold learning process as much as possible. If the supervised information cannot react in some situations (the pose angles between two points are same or similar), we can use the feature-based distance to construct the neighborhood of the manifold. Therefore, the linear representation of \(\tilde{d}_{ij}'\) (see Eq. (3)) is not the most optimized. As shown in Eq. (4), the parameter \(n\) is proportional to the influence of the supervised information. Our proposed redefinition of inter-point distance (two-grade comparison structure) can be illustrated as the situation that the parameter \(n\) approaches to the positive infinite.

3.2. Pose-constrained graph weight

The graph weights do not necessarily have a monotonic relationship with the degree of closeness of the related neighbors in the neighborhood. Therefore, we cannot directly impose a pose similarity function to the original graph weights in some manifold learning algorithms, such as LLE and NPE. In this section, we present two ways (direct and indirect) to incorporate the supervised information into the computation of the graph weight.

(1) For LE (nonlinear version) and LPP (linear version), we can adopt Heat Kernel (parameter \(t \in R\)) to determine the weights \(w_{ij}\), as follows:

\[
w_{ij} = e^{-\frac{||x_i - x_j||^2}{t}}
\]

We can see that \(w_{ij} > 0\) and the weight is proportional to the degree of closeness of the related neighbor. Therefore, we can directly apply a pose similarity function (like the function \(f([z_i - z_j])\) in Eq. (3)) to the weight. The adjusted graph weight \(\tilde{w}_{ij}\) can be described as:

\[
\tilde{w}_{ij} = e^{-\frac{f([z_i - z_j])}{t} \cdot ||x_i - x_j||^2} \quad (m > 1)
\]

(2) For LLE (nonlinear version) and NPE (linear version), the graph weight is calculated by the cost function (see Eq. (7)) rather than defined in a straightforward way, as in Eq. (5). The weights do not have a monotonic relationship with the degree of closeness of the related neighbors. Consider a reference data point \(\tilde{x}\) with \(K\) nearest neighbors \(\tilde{x}_i\) and reconstruction weights \(w_i\) that sum to one. The cost function (reconstruction error) \(\epsilon\) can be written as:

\[
\epsilon = ||\tilde{x} - \sum_i w_i \tilde{x}_i||^2 = \sum_i w_i \tilde{x}_i^2 - 2 \sum_i w_i \tilde{x}_i \cdot \tilde{x} + \tilde{x}^2
\]

where \(C_{ij}\) is the element of the local covariance matrix \(C\), which is symmetric and semipositive. The element \(C_{ij}\) is inversely proportional to the closeness degree of the related neighbors in the neighborhood. We can impose a pose similarity related function \(g_{ij}\) to \(C_{ij}\), which has the form:

\[
\tilde{C}_{ij} = g_{ij} \cdot C_{ij}
\]

where

\[
g_{ij} = \frac{|x_i - x_j|^n}{(\max(|x_i - x_j| - |x_i - x_j| + \varepsilon))^m}
\]

where \(z\) denotes the pose angle information (e.g., yaw pose degree) of the reference data point \(\tilde{x}\).

The new cost function will be:

\[
\tilde{\epsilon} = \sum_i w_i \tilde{x}_i \tilde{C}_{ij}
\]

which can be minimized in a closed form, using a Lagrange multiplier to enforce the constraint that \(\sum_j w_j = 1\), by solving the linear system of equations:

\[
\begin{cases}
w_j \sum_i \tilde{C}_{ij} = 1 \\
\sum_j w_j = 1
\end{cases}
\]

This way, the supervised information can be indirectly
incorporated into the graph weights.

4. Experiment and results

In this paper, we compare our proposed methods with the original and BME [9] methods on LLE, NPE, LE and LPP. Our methods contain (1) head pose estimation based on manifold learning with the proposed pose-constrained inter-point distance (Method 1), (2) head pose estimation based on manifold learning with the proposed pose-constrained inter-point distance and graph weight (Method 2), and (3) Method 2 with LEOH feature (Method 3). The original method corresponds to the manifold learning process without any supervised information and with grayscale pixel intensity based feature.

We adopt the Generalized Regression Neural Networks (GRNN) and Multivariate Linear Regression (MLR) to learn the mapping and regression model of manifold learning. We use error mean \( u \) and standard deviation \( \sigma \) to represent the estimating accuracy of the head poses. The expression \( |r - e| \) defines the estimation error where \( r \) is the real pose angle and \( e \) is the estimated pose angle. Then, \( u = E(|r - e|) \) where \( E(\cdot) \) represents the expectation and \( \sigma = \sqrt{E[(r - e) - u]^2} \).

4.1. Facial databases

In the experiments, we use two face databases: (1) FacePix [17] (various poses with constant illumination) and (2) MIT-CBCL [18] (various poses with varying illuminations).

The FacePix database contains 30 individuals with yaw pose angles varying from -90° to 90° in increments of 1°. Some examples are shown in Figure 2 (a). The resolution of the face images is 128×128, which is down-sampled to 48×48 in the experiments. These facial images have the same illumination.

![Figure 2. Examples of (a) FacePix and (b) MIT-CBCL.](image)

The “training-synthetic” set of MIT-CBCL face database contains 10 individuals, which are rendered from 3D head models. The yaw pose angles vary from -32° to 0° in increments of 4° with 36 kinds of illuminations. To extend the range of the poses, we evenly divide the facial images with the same pose into two parts and horizontally flip the images of one part, in which the illuminations are always arbitrarily distributed. Therefore, the yaw pose angles of the revised database vary from -32° to 32° in increments of 4° and each pose has 18 arbitrary illuminations. Some examples are shown in Figure 2 (b). The resolution of the face images is 200×200, which is also down-sampled to 48×48 in the experiments.

For both databases, we use 10-fold cross-validation. In addition, we use the original face images rather than the face-centered and cropped images with little background information [8, 11] for test.

4.2. Comparison between BME and Method 1

In this section, we compare two methods (BME [9] and Method 1) using LLE, each adopts its own pose-constrained inter-point distance. We calculate the error mean \( (u) \) with increasing number of nearest neighbors \( (K) \) on FacePix database, shown on Figure 3. The dimensional \( d \) of embedding space is fixed at 20.

![Figure 3. Comparison between BME and Method 1](image)

We can see that Method 1 performs better than the BME method when \( K \in (30,80) \). In addition, due to the orderliness of selected nearest neighbors of Method 1, the monotonicity of its result is better than that of BME in the same region of \( K \), such as \( 30 < K < 60 \) or \( 60 < K < 80 \). If \( K \) is too small, the difference between the two methods is slight since their nearest neighbors only include the data points with the same pose. If \( K \) is too big, neither method can represent the local geometry of the neighborhood.

4.3. Visualization

In this section, we compare the difference of the four methods (original, BME, Method 2 and Method 3) by visualizing their manifolds in 3D embedding space. We take NPE on FacePix (use 27 individuals with \( K = 50 \) and \( d = 3 \)) as an example, shown on Figure 4.

Due to the linear projection of NPE, these manifolds are not smooth. Compared with the original and BME methods, the manifold of Method 2 has a compact pose distribution and is robust to identity variation. For Method 3, the generalization is the best of them.

4.4. Head pose estimation without illumination variation

In this section, we compare the estimating accuracy \( (u \pm \sigma) \) of the four methods with \( K = 50 \) and \( d = 20 \) on FacePix with the constant illumination, shown on Table 1.

As shown in Table 1, the performance of Method 3 is the
best of all the four methods; the performance of Method 2 is also better than that of the original and BME method. The comparison of the experimental results proves that the proposed pose-constrained variables (inter-point distance and graph weight) and LEOH feature can both improve the performance of head pose estimation and are robust to the identity variation.

4.5. Head pose estimation with illumination variation

In this section, we compare the estimating accuracy ($u \pm \sigma$) of the four methods with $K = 200$ and $d = 20$ on MIT-CBCL with different illuminations, shown on Table 2.

We can see that their results are all fairly good because the faces of the database are all centered in the image with black background (almost without hair, see Figure 2 (b)). In addition, for the same pose, there are many training samples (9 individuals × 18 illuminations) even though with different illuminations. For the performance comparison, Method 3 is also the best and Method 2 takes the second place. Besides the identity-robust, they both show a good illumination-robust ability.

5. Conclusion

We have proposed pose-constrained variables (inter-point distance and graph weight) and LEOH feature for manifold learning. The main novelty of our work is that, compared to previous work, we incorporated the supervised information into the inter-point distance and graph weight which affect the property of the manifold, and used localized edge-based robust feature rather than pixel intensity feature for manifold learning. Experiments showed that the proposed methods are both effective to improve the estimating accuracy of the head poses under varying identities and illuminations.

6. References


Table 1. Comparison of the original, BME, Method 2, Method 3 on FacePix.

<table>
<thead>
<tr>
<th></th>
<th>LLE</th>
<th>NPE</th>
<th>LE</th>
<th>LPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>BME</td>
<td>7.8367±5.6229</td>
<td>8.6417±6.5936</td>
<td>6.1297±5.3732</td>
<td>6.0294±5.4668</td>
</tr>
<tr>
<td>Method 2</td>
<td>6.0707±5.4541</td>
<td>6.0021±5.5254</td>
<td>5.6092±4.8949</td>
<td>5.7994±5.0865</td>
</tr>
</tbody>
</table>

Table 2. Comparison of the original, BME, Method 2, Method 3 on MIT-CBCL.

<table>
<thead>
<tr>
<th></th>
<th>LLE</th>
<th>NPE</th>
<th>LE</th>
<th>LPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>4.6789±3.7486</td>
<td>5.0017±4.2487</td>
<td>2.0418±1.2679</td>
<td>2.9936±1.8250</td>
</tr>
<tr>
<td>BME</td>
<td>3.2803±2.3316</td>
<td>3.8915±3.2684</td>
<td>1.8364±1.4521</td>
<td>2.0847±1.7492</td>
</tr>
<tr>
<td>Method 2</td>
<td>1.8067±1.4470</td>
<td>2.1891±1.9074</td>
<td>1.5102±1.2475</td>
<td>1.6083±1.2085</td>
</tr>
<tr>
<td>Method 3</td>
<td>1.2175±0.8496</td>
<td>1.3731±1.0728</td>
<td>1.2809±0.9035</td>
<td>1.3301±1.1108</td>
</tr>
</tbody>
</table>