Robust Head Pose Estimation Using Supervised Manifold Learning

Chiraz BenAbdelkader

New York Institute of Technology,
Abu Dhabi, United Arab Emirates
cbenabde@nyit.edu

Abstract. We address the problem of fine-grain head pose angle estimation from a single 2D face image as a continuous regression problem. Currently the state of the art, and a promising line of research, on head pose estimation seems to be that of nonlinear manifold embedding techniques, which learn an "optimal" low-dimensional manifold that models the nonlinear and continuous variation of face appearance with pose angle. Furthermore, *supervised* manifold learning techniques attempt to achieve this robustly in the presence of latent variables in the training set (especially identity, illumination, and facial expression), by incorporating head pose angle information accompanying the training samples. Most of these techniques are designed with the classification scenario in mind, however, and are not directly applicable to the regression scenario where continuous numeric values (pose angles), rather than class labels (discrete poses), are available. In this paper, we propose to deal with the regression case in a principled way. We present a taxonomy of methods for incorporating continuous pose angle information into one or more stages of the manifold learning process, and discuss its implementation for Neighborhood Preserving Embedding (NPE) and Locality Preserving Projection (LPP). Experiments are carried out on a face dataset containing significant identity and illumination variations, and the results show that our regression-based approach far outperforms previous supervised manifold learning methods for head pose estimation.

Keywords: head pose estimation, supervised learning, manifold learning, dimensionality reduction, nonlinear regression.

1 Introduction

Head pose estimation from a single 2D image is a basic and important task of many face processing applications, viz. face recognition, face and person tracking, and human-machine interfaces [1,2,3,4]. In face recognition systems, head pose is a major source of (obviously unwanted) intra-person facial appearance variability, which can be removed by performing head pose estimation as a preprocessing step to select only face images with similar head poses for face matching. In human-computer interfaces, head pose provides a strong cue for determining a person's gaze direction and thereby inferring their focus of attention, intent,

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 518–531, 2010.
© Springer-Verlag Berlin Heidelberg 2010

and behavior. Pose estimation can also be used as a front end processing module for face tracking to bootstrap the tracker and re-initialize it when it drifts off.

Previous work on head pose estimation from 2D images can be divided across several categories: coarse (discrete) vs. fine-grain (continuous), geometric-based vs. appearance-based methods, holistic vs. local region based. The interested reader is referred to the recent surveys for a comprehensive review [5,3].

Currently, the state of the art on head pose estimation, and a promising line of research, seems to be in manifold embedding, a special class of dimensionality reduction techniques that attempt to learn a low-dimensional manifold on which the data lies [6,7,8]. A fundamental underlying assumption of this approach is that face images with varying head pose are —geometrically speaking— points that reside on or near a low-dimensional manifold embedded in the ambient high-dimensional input space (image space), and whose intrinsic dimensionality is no more than the number of degrees of freedom of head movement [3]. This (pose) manifold models the nonlinear and continuous variations of face appearance with pose angle, and *if* learned properly, can be used to accurately predict pose angle from face images. But this manifold is highly nonlinear and complex and learning it is no easy task, particularly in the presence of distracting variation in the dataset, namely background clutter, natural variations (identity, facial expression), and imaging variations (illumination, blur, noise, etc.) in the face images.

As in any statistical learning problem, a necessary condition for accurate learning to take place is to somehow suppress extraneous variables in the training set while preserving variables of interest (the pose variable in our case). This is a recurring problem in the face processing literature, and is generally handled using one or a combination of the following two general approaches:

Image Preprocessing: preprocess the face images to extract and/or enhance certain low-level features such as histogram equalization, edge enhancement, Gabor wavelets, histogram of gradients (HOG), etc. This approach can work well for suppressing certain imaging variations, misalignment errors, and background variations.

Supervised Learning: use auxiliary information associated with the training set to bias the learning process. This approach is widely used in classification scenarios, with auxiliary information consisting of class labels of the variable of interest (e.g. subject labels in the case of face recognition).

The focus of this paper is on developing manifold learning methods of the second category in the context of head pose estimation. Previous research in this area has, for the most part, viewed the problem as a classification problem wherein the viewing sphere is (artificially) quantized into non-overlapping subintervals, and head pose is represented by a set of discrete pose labels–rather than a continuum of pose angles. This approach appears to be adequate for *coarse* pose estimation (with some reservations) [9,10,11,12,13,14,15], and other classification problems such as facial expression and face recognition [16,17,18,19,20]. It is, however, fundamentally flawed when used for fine-grain pose estimation for two

main reasons: (i) pose estimation discontinuities occur at class boundaries due to the arbitrary nature of the pose classes, (ii) the numerical properties (scale, well-ordering) of the underlying pose angles are lost; for example, the difference between pose label 1 and pose label 2 is viewed no differently than between pose labels 1 and 5.

To date relatively little work exists that attempts to solve head pose estimation as a regression problem proper within a nonlinear manifold learning framework [21,22,23]. In this paper we present a principled and detailed look into this approach. Specifically, we propose a *taxonomy* of methods for using pose angles associated with the training set in the various stages of the manifold learning process. We demonstrate the proposed techniques on Neighborhood Preserving Embedding (NPE) [24] and Locality Preserving Projection (LPP) [25,17], which are linearized versions of the well-known manifold learning methods locally linear embedding (LLE) and Laplacian eigenmaps (LE), respectively. Experimental results on the FacePix database [26] show that our regression-based approach is robust to identity and illumination variations, and clearly outperforms recent similar pose estimation methods such as [11,23,14].

The remainder of the paper is organized as follows. Section 2 gives an overview of manifold embedding techniques using graph embedding as a general framework. Section 3 presents our taxonomy of supervised manifold learning methods. Section 4.4 gives experiments and results on the FacePix and AT&T datasets. Finally, Section 5 concludes with a summary and directions for future work.

2 A General Framework for Manifold Learning

Manifold learning algorithms can in general be cast in terms of a graph embedding problem based on a specific intrinsic graph that encodes certain desired statistical or geometric properties of the dataset [27]. Specifically, given a dataset of n points in p-dimensional space (denoted $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$), a manifold learning algorithm basically executes a four-stage pipeline of the following form:

1. *Neighborhood computation.*
   For each data point $\mathbf{x}_i$, determine its k closest points (called *neighborhood*), where k is a design parameter and proximity/nearness is based on some inter-point distance metric, $\mathbf{D}_{ij}$, such as Euclidean, geodesic, Mahalanobis, cosine, etc. In our case, points are face images and hence inter-point distances represent appearance dissimilarity between face images.
2. *Neighborhood graph construction.*
   A weighted graph G is constructed whose vertices are the data points, edges connect each point with its neighbors (as defined in the previous step), and the weight of an edge, $\mathbf{W}_{ij}$, represents some measure of *affinity* or similarity between two neighbor points. Intuitively, this graph encodes the intrinsic local geometry of the manifold from which the data set is sampled.

3. *Computation of a low-dimensional graph embedding.*
   This seeks a set of n d-dimensional vectors, $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n$, that preserves the properties of graph G, in other words that preserves the intrinsic geometry of the underlying manifold. This task reduces to the optimization of a quadratic form under proper regularization constraints (typically scale normalization to avoid the trivial solution), and has a closed-form solution consisting of the smallest eigenvectors of $\mathbf{B^{-1}L}$, where $\mathbf{B}$ is a diagonal matrix such that B_{ii} is equal to the sum of the ith row of $\mathbf{W}$ and $\mathbf{L} = \mathbf{B} - \mathbf{W}$ is the Laplacian of graph G.
4. *Computation of an input-to-embedding mapping.*
   This seeks a mapping that transforms new (out-of-sample) points in the p-dimensional input space to d-dimensional space, which can be solved as a non-linear regression problem on the embedding vectors obtained in the previous step, for example using GRNN's or support vector machines [23]. But clearly this step is only required where prediction (rather than visualization) is of interest.
   Because non-linear regression in a high-dimensional space is itself tricky, some techniques bypass it and instead constrain the mapping to be linear, effectively combining the third and fourth stages into one computational step. This amounts to finding the best d-dimensional linear subspace *approximation* for the nonlinear manifold. Examples of "linearized" techniques notably include Locality Preserving Projections (LPP) [25,17] which is an extension of Laplacian Eigenmaps [28], and Neighborhood Preserving Embedding (NPE) [24] and Locally Embedded Analysis (LEA) [11], both of which are linearized variants of Locally Linear Embedding [29,6].

3 Our Taxonomy of Supervised Manifold Learning

In the context of classification, supervised manifold learning generally aims to find a low-dimensional space that maximizes the separation of points from different classes while minimizing that of points within the same class (between- and within- class scatter, respectively). However in the regression scenario, the goal is rather to find directions (axes) that best *predict* the regression variable(s) associated with the dataset, in our case, the head pose angles.

With the general four-stage manifold learning framework of Section 2 in mind, we propose a *taxonomy* of methods that correspond to different ways of incorporating the pose angle information (denoted as $z_1, z_2, \cdots, z_n$) at the different stages. Our taxonomy represents a more comprehensive treatment of the regression scenario than any previous work on head pose estimation [22,23].

3.1 Overview

A summary of the taxonomy follows below, and Table 1 compares the proposed methods with previous work on supervised manifold learning, both in the context of classification and regression scenario. Clearly, the latter remains mostly wide open for contributions, which is the goal of this work.

Stage 1: Option 1.1 Construct neighborhoods using as proximity metric the similarity of z values, i.e. the neighborhood of a sample $\mathbf{x}_i$ consists of the k data samples whose z values are most similar to z_i.

Option 1.2 Construct neighborhoods using as proximity metric the inter-point distances adjusted according to the dissimilarity of respective z values.

Stage 2: Option 2.1 Adjust the graph weights (matrix $\mathbf{W}$) based on the similarity of respective z values.

Stage 3: Option 3.1 Incorporate regression information as an additional term in the objective function to be optimized.

Option 3.2 Incorporate regression information as additional constraints in the function optimization.

Table 1. Previous supervised manifold learning techniques that are related to our proposed taxonomy, in the classification (C) and regression (R) scenarios, used for different applications (face recognition (FR), pose estimation (PE), face expression recognition (FE), visualization (V), and other (O))

	[19]	[24]	[11]	[30]	[20]	[31]	[10]	[32]	[23]	[18]	[13]	[14]	[15]
Scenario	C	C	C	C	C	C	C	C	R	C	C	C	C
Application	FE	FR	V,PE	O	FR	O	O	O	PE	FR	V	PE	FR
Related to 1.1	x	x	x	x	x								
Related to 1.2						x	x	x	x				
Related to 2.1													
Related to 3.1													
Related to 3.2										x	x	x	x

3.2 Supervised Stage 1

Ideally, in order for accurate manifold learning to occur, one needs to capture the *true* neighborhood structure of the data set in the underlying manifold. However, the actual neighborhood of a data sample in the ambient input space may be contaminated with "fake" neighbors due to the presence of noise, confounding factors, and sparse sampling. The above taxonomy contains two different ways of exploiting regression information to more robustly distinguish *true* neighbors and filter out *fake* ones. Option 1.1 relies exclusively on this information while Option 1.2 attempts to reconcile information from both the inter-point distances and regression value similarities.

In principle, Option 1.2 can be implemented in infinitely many ways by using different penalty functions spanning the entire gamut between the two extremes: using only inter-point distances (the unsupervised option) and using only regression information (Option 1.1). For example in [23] Balasubramanian et al. have suggested a family of functions of the form:

$$\widetilde{D}_{ij} = f(|z_i - z_j|) \cdot D_{ij} \tag{1}$$

where $\mathbf{D}$ is the original inter-point distance matrix, $\widetilde{\mathbf{D}}$ is the adjusted distance matrix, and f is some reciprocal *increasing* positive function. We currently use the following reciprocal function: $f(u) = \alpha \cdot u/(\beta - u)$ where α and β are scalar parameters. The choice of a reciprocal function seems appropriate because it ensures that the penalty increases at a faster rate at larger values of $|z_i - z_j|$. An exponential function might also work for this same reason.

The relative merits of Option 1.1 and Option 1.2 in effect depend on the sampling density and geometric structure of the underlying manifold. Also, using the ϵ-ball approach rather than the k nearest neighbors approach may be more helpful in some cases.

Both methods are closely related to previous supervised manifold learning techniques developed for the classification scenario. Specifically, Option 1.1 is akin to techniques that limit the neighborhood to points of the *same* class [19,11,30,20]. Interestingly, Teoh et al. call this approach "neighborhood discriminant criterion" and argue that it is equivalent to the Fisher discriminant criterion [20]. Option 1.2 is akin to methods that adjust the inter-point distances by reducing those of same-class point pairs and/or penalizing those of point pairs of different classes [31,10,32].

3.3 Supervised Stage 2

Recall that graph weights $\mathbf{W}$ represent the geometric structure of the local neighborhood based on some inter-point distance measure. Furthermore, W_{ij} essentially determines the contribution of neighbor pair $\mathbf{x}_i$ and $\mathbf{x}_j$ in the computation of the optimal embedding in Stage 3. But because neighborhoods may be contaminated with "fake" neighbors that distort the embedding, and just as we have used regression information in Stage 1 to determine the neighborhoods more robustly (Section 3.2), we can similarly use it to penalize (i.e. reduce) the contribution of a neighbor pair by a factor proportional to the dissimilarity between their respective regression values. In other words, the new (adjusted) graph weights could of the form:

$$\widetilde{W}_{ij} = W_{ij} \cdot g(|z_i - z_j|) \tag{2}$$

or it could also be of the form:

$$\widetilde{W}_{ij} = W_{ij} + g(|z_i - z_j|) \tag{3}$$

where $g(u)$ is some positive decreasing function, such as a negative exponential (Gaussian kernel) or a reciprocal. Interestingly, using the second (additive) form is actually equivalent to adding a term to the objective function in Stage 3, hence equivalent to Option 3.1 (Section 3.4).

3.4 Supervised Stage 3

Recall that the objective function optimization represents preserving certain intrinsic geometric properties of the manifold. Hence we can incorporate regression information at this stage by extending the objective function with an additive term that represents some other geometric property to be preserved (Option 3.1). Alternatively, or simultaneously, we can incorporate this information in the form of constraints that represent some condition or property that should be avoided (Option 3.2).

Interestingly, Local Fisher Discriminant Analysis (LFDA) [12,13] and Local Discriminant Analysis (LDE) [18] both represent possible implementations of our Option 3.2 concept, *though* they are limited to the classification scenario. In LFDA, the objective function and constraints consist of "localized" versions of within- and between- class scatter, respectively. LDE uses a very similar idea. Also "kernelized" variants of both these methods were implemented using the kernel trick.

Note, however, that in order *not* to forego the convenience of solving the problem in closed-form as a generalized eigenvalue problem ($\mathbf{Ay} = \lambda\mathbf{By}$), both the additional objective function term and the constraints need to be expressed as a positive definite quadratic form ($\mathbf{y^T \Gamma y}$ where $\mathbf{\Gamma}$ is positive definite). Also, the new constraints should replace (and ideally *supersede*) the original constraints of the unsupervised method because there is no room for using both. We *could* use a non quadratic objective function and more than one set of non-quadratic constraints, but at the price of giving up the convenience of a closed-form solution for an iterative slower solution.

Below we discuss possible implementations of Option 3.1 in the context of two specific manifold learning methods: Locally Linear Embedding (LLE) and Laplacian Eigenmaps (LE). Extension to their linearized versions (NPE and LPP) is trivial. Possible implementations of Option 3.2 is work in progress, but suffice it to note here that a viable approach is to extend or generalize the LFDA concept of using between-class scatter to the regression scenario.

Implementation for LLE. We modify the usual LLE objective function by adding a second term [29,6] :

$$\Phi_{\mathrm{LLE}}(\mathbf{y}) = \sum_i |y_i - \sum_j \Omega_{ij} y_j|^2 + \lambda \, \frac{1}{2} \sum_{i,j} (y_i - y_j)^2 \Lambda_{ij} \tag{4}$$

where λ is a scalar constant and $\mathbf{\Omega}$ is the nxn reconstruction weights matrix, and $\mathbf{\Lambda}$ is a nxn matrix that represents some measure of similarity between the $\mathbf{z}$ values of neighbor points. Clearly the intuition is to *simultaneously* (i) preserve the local neighborhood structure, and (ii) keep neighbor points with more similar pose angles closer. However note that how well this works out is closely tied to the supervision methods of Stages 1 and 2 (Sections 3.2 and 3.3), since they determine the neighborhood and the contribution weight of each neighbor pair.

It is easy to show that Equation (4) reduces to :

$$\Phi_{\mathrm{LLE}}(\mathbf{y}) = \mathbf{y}^T\mathbf{M}\mathbf{y} + \lambda\mathbf{y}^T\widetilde{\mathbf{L}}\mathbf{y} = \mathbf{y}^T(\mathbf{M} + \lambda\widetilde{\mathbf{L}})\mathbf{y} \tag{5}$$

where $\mathbf{M} = (\mathbf{I}-\mathbf{\Omega})^T(\mathbf{I}-\mathbf{\Omega})$ and $\widetilde{\mathbf{L}}$ is the Laplacian of the affinity graph induced by $\mathbf{\Lambda}$. Clearly the extension to linearized versions of LLE (such as NPE and LEA) is trivial, as we have merely replaced $\mathbf{M}$ with $\mathbf{M} + \lambda\widetilde{\mathbf{L}}$.

We currently define similarity matrix $\mathbf{\Lambda}$ based on the heat kernel function as follows, though in principle other *decreasing* functions of $|z_i - z_j|$ would do:

$$\Lambda_{ij} = \begin{cases} \exp(-\frac{1}{2}|z_i - z_j|^2/\sigma^2) & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors } \textit{and } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where σ is a design parameter (the Gaussian kernel width).

Implementation for LE. We modify the usual LE objective function by adding a second term [28] :

$$\Phi_{\mathrm{LE}}(\mathbf{y}) = \frac{1}{2}\sum_{i,j}(y_i - y_j)^2 W_{ij} + \lambda\,\frac{1}{2}\sum_{i,j}(y_i - y_j)^2\Lambda_{ij} \tag{7}$$

$$= \frac{1}{2}\sum_{i,j}(y_i - y_j)^2(W_{ij} + \lambda\,\Lambda_{ij}) \tag{8}$$

$$= \mathbf{y}^T(\mathbf{L} + \lambda\,\widetilde{\mathbf{L}})\mathbf{y} \tag{9}$$

where λ is a scalar constant and $\mathbf{\Lambda}$ and $\widetilde{\mathbf{L}}$ are as defined above in Section 3.4. Again, the extension to linearized versions of LE (such as LPP) is trivial as we have merely replaced $\mathbf{L}$ with $\mathbf{L} + \lambda\widetilde{\mathbf{L}}$.

3.5 Discussion

A common thread runs through all these methods we have proposed: to *highlight* pose variations and *suppress* variations due to other (extraneous) factors. Specifically, given the local neighborhood nature of nonlinear manifold learning, we propose to achieve this by using the regression information to: (i) determine the neighborhood (Stage 1), (ii) determine the contribution of each neighbor pair (Stage 2), and (iii) define new or additional manifold structure preservation properties (Stage 3). These methods are complementary to some extent (at least not entirely redundant) and can certainly be used in tandem. However, the inner workings of each method depends both on the dataset: how much variation it contains and how sparsely sampled the pose angles are. Also, because these methods are so closely related, it is not clear how the synergy between them will affect performance when they are used together. Further analytical and empirical work is needed to study this synergy.

Fig. 1. Sample images from the FacePix database

4 Experiments and Results

4.1 The Data

Manifold learning and pose estimation are challenging tasks when the input face images contain significant variation, such as from illumination and identity. We test our proposed taxonomy on the FacePix database which contains face images of 30 subjects with both pose and illumination variations, namely:

Variable pose, constant illumination: 181 images for each subject captured with the yaw head pose angle varying at 1-degree increments in the range [-90,90], and with constant ambient illumination.

Constant pose, variable illumination: 181 images for each subject captured with the yaw illumination angle varying at 1-degree increments in the range [-90,90], and with constant frontal pose. Furthermore, this is done with two different illumination intensities: dark and light.

The pose and illumination angles associated with each image were annotated using a precisely calibrated mechanism. Also, the face images are scaled and aligned such that the eyes, nose, and mouth remain at fixed pixel positions in each image [26,23]. A sample of these images is shown in Figure 1. For the purpose of our experiments, we have applied some more preprocessing on these images by cropping the middle 98x98 rectangle to remove some of the background and shoulder areas, and then downsampling to a size of 25x25 pixels.

4.2 Methodology

We summarize our validation methodology in the following points:

- Use two different manifold learning algorithms: NPE [24], LPP [25].
- Use different supervision modes based on combining different implementation options for Stage 1 and Stage 3. We do not incorporate supervision into Stage 2 because it is somewhat equivalent to Stage 3 (as noted earlier).

- Use two different regression methods to estimate head pose angle from embedded face images: support vector regression (SVR) with Gaussian RBF kernel and smoothing cubic splines.
- Test on three subsets of the FacePix face images: (i) images with 1-degree pose angle increments, (ii) images with 10-degree pose angle increments, and (iii) subset (i) plus images with frontal pose and 1-degree illumination angle increments.
- Use leave-one-out cross validation to estimate pose estimation error (whereby images of 29 subjects are used for training and the images of the remaining subject are used for testing).

4.3 Visualization

Figure 2 shows the 3-dimensional embedding of the face images of 20 subjects from the FacePix dataset, based on four different manifold learning techniques and combination of supervision methods Option 1.1 and Option 3.1. In general, all methods yield a one-dimensional manifold (as expected) that is more or less ordered by pose angle, at least visually speaking. The LLE and LE manifolds are quite compact and smooth; NPE's manifold is less smooth; and LPP's manifold is the least smooth. The fact that NPE and LPP's embeddings are not as smooth as those obtained by LLE and LE is not surprising, since they only seek a linear subspace approximation of the embedding.

To get a better sense of how pose angle varies varies along these pose manifolds, we analyze the identity and pose of the (Euclidean) neighbors of each data point in the 3-dimensional embedding. Figure 3 shows that, as desired, overall each data point is surrounded by points of the same pose rather than points of the same identity. However, again, this trend is better exhibited in the LLE and LE manifolds than those of NPE and LPP.

4.4 Pose Estimation Results

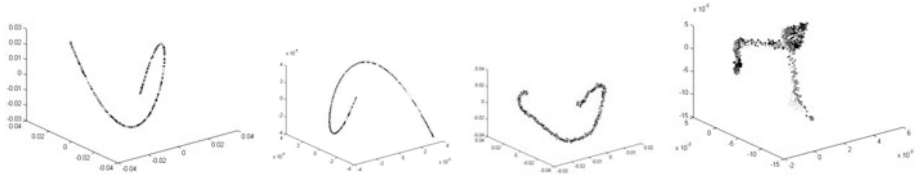
Table 2 and Table 3 show the pose angle estimation error results when using Support Vector Regression and splines, respectively, for estimating pose from the embedded face images, with $d = 20$ and $k = 50$. These results basically compare the performance of different manifold learning methods with different supervision modes. Clearly, the best performance is achieved with NPE and with the last two supervision modes, wherein supervision is incorporated both in Stages 1 and 3. The second supervision option for Stage 1 (i.e. Option 1.2) seems to perform significantly better than the first one. Overall, NPE performs better than LPP and spline regression better than support vector regression. Also, interestingly performance for the dataset containing both illumination variation and pose variation is not behind that of the dataset containing pose variation only. Hence, based on these results and on Table 1, our proposed supervision methods are superior to previous work such as [23,14].

Table 2. Mean absolute deviation of the pose angle error (in degrees), using support vector regression for pose estimation

	1-deg Pose variation		*10-deg Pose variation*		*Pose+Illum. variation*	
	NPE	LPP	NPE	LPP	NPE	LPP
unsupervised,unsupervised	8.2	9.5	11.2	14.1	9.1	15.6
Option 1.1,unsupervised	6.0	8.1	10.6	12.5	8.3	10.1
Option 1.2,unsupervised	5.5	7.9	10.8	10.3	7.7	10.0
unsupervised,Option 3.1	5.2	6.8	7.3	7.9	5.3	7.7
Option 1.1,Option 3.1	4.4	5.2	5.0	6.7	4.3	5.5
Option 1.2,Option 3.1	4.5	5.0	5.1	6.7	4.7	4.9

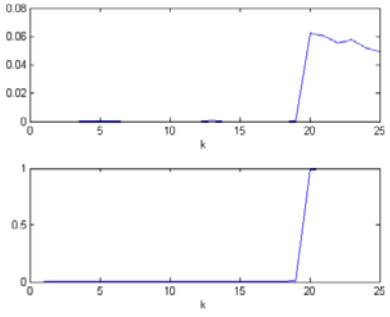
Table 3. Mean absolute deviation of the pose angle error (in degrees), using smoothing cubic splines for pose estimation

	1-deg Pose variation		*10-deg Pose variation*		*Pose+Illum. variation*	
	NPE	LPP	NPE	LPP	NPE	LPP
unsupervised,unsupervised	5.2	8.2	9.6	12.1	8.6	13.4
Option 1.1,unsupervised	4.2	7.0	9.1	10.5	7.3	9.7
Option 1.2,unsupervised	4.3	6.6	8.0	9.2	7.0	8.2
unsupervised,Option 3.1	3.5	4.6	4.6	6.1	3.9	4.3
Option 1.1,Option 3.1	2.1	3.2	4.7	5.9	3.6	4.4
Option 1.2,Option 3.1	**1.5**	3.4	3.5	5.2	2.6	3.5

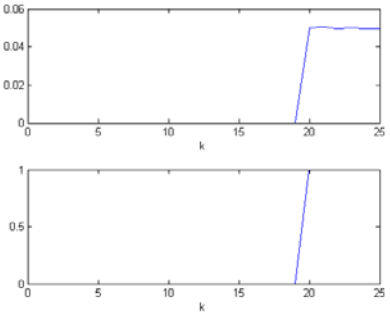


(a) (b) (c) (d)

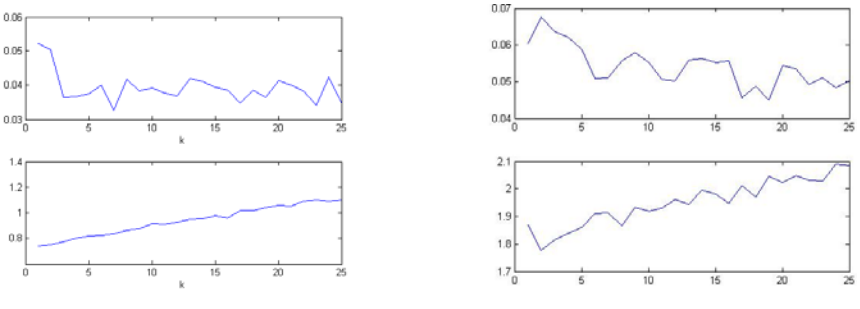
Fig. 2. 3-dimensional embedding of face images of 20 subjects of the FacePix dataset, using $k = 25$ and four different manifold learning techniques: (a) supervised LLE, (b) supervised LE, (c) supervised NPE, (d) supervised LPP. The data points are color coded differently for each subject label.



(a)



(b)



(c) (d)

Fig. 3. Neighborhood analysis in 3-dimensional embedded space based on (a) supervised LLE, (b) supervised LE, (c) supervised NPE, (d) supervised LPP. Top figure plots probability that kth NN is of same subject versus k, and bottom figure plots absolute mean deviation of kth nearest neighbor's pose angle versus k.

5 Conclusions and Future Work

We have proposed a taxonomy of methods for solving pose estimation as a proper (continuous) regression problem within the general nonlinear manifold learning framework. The main novelty of our work lies in that, compared to previous work, we take a more comprehensive approach to the way we exploit supervision information (pose angles) into the learning process. Experiments on a face dataset containing significant identity and illumination variation have shown that our methods significantly outperform related recent work such as [11,23,14]. However, there is undoubtedly great room for improvement, most notably:

- Further analytical and empirical work to characterize the relationships between the supervision methods of the different stages, particularly in relation to pose estimation (regression) performance.
- Test on other manifold learning techniques such as Isomap [33].
- Extend to kernelized versions of NPE and LPP [27], as they only give the best linear subspace approximation of the low-dimensional embedding.
- Apply some clever feature extraction and/or preprocessing techniques on the face images (such as Gabor wavelets, histogram of gradients) to remove unwanted variation, to simplify the manifold learning process.
- Test on benchmark datasets containing more pose+illumincation variations.
- Test on benchmark datasets containing more challenging variations (facial expression, face alignment errors).

References

1. Tian, Y., Brown, L., Connell, J.: Absolute head pose estimation from overhead wide-angle cameras. In: IEEE Workshop on Analysis and Modeling of Faces and Gestures (2003)
2. Wu, J., Trivedi, M.M.: A two-stage pose estimation framework and evaluation. Pattern Recognition 41, 1138–1158 (2008)
3. Murphy-Chutorian, E., Trivedi, M.: Head pose estimation in computer vision: A survey. 31, 607–626 (2009)
4. Orozco, J., Gong, S.: Head pose classification in crowded scenes. In: British Machine Vision Conference (2009)
5. Brown, L., Tian, Y.: Comparative study of coarse head pose estimation. In: IEEE Workshop on Motion and Video Processing (2002)
6. Saul, L.K., Roweis, S.T.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. Journal of Machine Learning Research 4, 119–155 (2003)
7. Burges, C.J.C.: Geometric Methods for Feature Extraction and Dimensional Reduction. In: Book, Kluwer Academic Publishers, Dordrecht (2005)
8. van der Maaten, L., Postma, E., van den Herik, H.: Dimensionality reduction: A comparative review. Technical Report TiCC-TR 2009-005, Tilburg University (2009)
9. Raytchev, B., Yoda, I., Sakaue, K.: Head pose estimation by nonlinear manifold learning. In: ICPR, pp. 462–466 (2004)
10. Geng, X., Chuan Zhan, D., Hua Zhou, Z.: Supervised nonlinear dimensionality reduction for visualization and classification. IEEE Transactions on Systems, Man, and Cybernetics: Part B 35, 1098–1107 (2005)
11. Fu, Y., Huang, T.S.: Graph embedded analysis for head pose estimation. In: FGR, pp. 3–8 (2006)

12. Sugiyama, M.: Local fisher discriminant analysis for supervised dimensionality reduction. In: International Conference on Machine learning, pp. 905–912 (2006)
13. Sugiyama, M.: Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. Journal of Machine Learning Research 8, 1027–1061 (2007)
14. Wang, X., Huang, X., Gao, J., Yang, R.: Illumination and person-insensitive head pose estimation using distance metric learning. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 624–637. Springer, Heidelberg (2008)
15. He, X., Ji, M., Bao, H.: Graph embedding with constraints. In: IJCAI, pp. 1065–1070 (2009)
16. Yang, M.H.: Extended isomap for pattern classification. In: ICPR (2002)
17. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.J.: Face recognition using laplacianfaces. PAMI 27, 328–340 (2005)
18. Chen, H.T., Chang, H.W., Liu, T.L.: Local discriminant embedding and its variants. In: CVPR, pp. 846–853 (2005)
19. Zhao, Q., Zhang, D., Lu, H.: Supervised LLE in ICA space for facial expression recognition. In: International Conference on Neural Networks and Brain, pp. 1970–1975 (2005)
20. Teoh, A.B.J., Pang, Y.H.: Analysis on supervised neighborhood preserving embedding. IEICE Electronics Express 6, 1631–1637 (2009)
21. Nilsson, J., Sha, F., Jordan, M.I.: Regression on manifolds using kernel dimension reduction. In: International Conference on Machine Learning, pp. 697–704 (2007)
22. Balasubramanian, V.N., Ye, J., Panchanathan, S.: Biased manifold embedding: a framework for person-independent head pose estimation. In: CVPR (2007)
23. Balasubramanian, V.N., Krishna, S., Panchanathan, S.: Person-independent head pose estimation using biased manifold embedding. Eurasip Journal on Advances in Signal Processing 2008, 1–15 (2008)
24. He, X., Cai, D., Yan, S., Zhang, H.J.: Neighborhood preserving embedding. In: ICCV, pp. 1208–1213 (2005)
25. He, X., Niyogi, P.: Locality preserving projections. Advances in Neural Information Processing Systems 16, 100–200 (2004)
26. Little, G., Krishna, S., Black, J., Panchanthan, S.: A methodology for evaluating robustness of face recognition algorithms with respect to variations in pose angle and illumination angle. In: International Conference on Acoustics, Speech, and Signal Processing (2005)
27. Yan, S., Xu, D., Zhang, B., Zhang, H.J., Yang, Q., Lin, S.: Graph embedding and extensions: A general framework for dimensionality reduction. PAMI 29, 40–51 (2007)
28. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Journal of Neural Computation 15, 1373–1396 (2003)
29. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
30. Zeng, X., Luo, S.: A supervised subspace learning algorithm: Supervised neighborhood preserving embedding. In: International Conference on Advanced Data Mining and Applications (2007)
31. de Ridder, D., Kouropteva, O., Okun, O., Pietikùinen, M., Duin, R.P.: Supervised locally linear embedding. In: International Conference on Artificial Neural Networks and Neural Information Processing (2003)
32. Li, C.G., Guo, J.: Supervised isomap with explicit mapping. In: Innovative Computing, Innovation, and Control (2006)
33. Tenenbaum, J.B., Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)