Pattern Anal Applic
DOI 10.1007/s10044-016-0557-8

THEORETICAL ADVANCES

A clustered locally linear approach on face manifolds for pose estimation

C. V. Hari[1] · Praveen Sankaran[1]

Received: 2 October 2015 / Accepted: 4 May 2016

Abstract Data points with small variations between them are assumed to lie close to each other on a smooth varying manifold in the feature space. Such data are hard to classify into separate classes . A sequence of face pose images with closely varying pose angles can be considered as such data. The pose angles when large enough create images that are largely differing from each other, and thus, the sequence of face images can be assumed to be on or near a nonlinear manifold. In this paper, we propose an unsupervised pose estimation method for face images based on clustered locally linear manifolds using discriminant analysis. We divide the data into multiple disjointed, locally linear and separable clusters. The problem of identifying which cluster to use is solved by dividing the entire process into two steps. The first step or projection using the entire smooth manifold identifies a rough region of interest. We use clustering techniques on entire data to form the pose-dependent classes which are then used to find the first set of discriminant functions. The second step or second projection uses trained cluster(s) from this neighbourhood to obtain a second set of discriminant functions. The idea behind such an approach is that the local neighbourhood would be linear and provide better between-class separation, and hence, the classification problem would now be simpler.

Keywords Pose estimation · Clustering · Smooth manifolds · Discriminant analysis · Multiple subspaces

✉ C. V. Hari
haricv@gmail.com

[1] Department of Electronics and Communication Engineering, National Institute of Technology Calicut, Kerala, India

1 Introduction

Face pose estimation methods try to identify the orientation of human face angles present in an image. Pose angle estimation from face images is a subproblem of most face analysis problems. Major research areas under pose estimation are human–computer interface (HCI), 3D face modelling, gaze identification, face expression recognition, face recognition. . .etc. The knowledge about pose direction can contribute useful information about a person's intent and behaviour. Also, there is an important meaning in the movement of the head as a form of gesturing in a conversation [1]. Next level of development of computer vision would be obtained through the combined effect of head pose estimation and gaze tracking [2].

In a pose estimation problem, the training classes should be pose dependent and should not be person dependent. Also, availability of maximum possible pose angles for training is key to identifying poses to a good resolution. Such a training database would result in nonlinear face pose manifolds in which the class separation between the data points is small. We set the following objectives in this work:

1. To develop an unsupervised method to obtain the discriminant functions.
2. To develop optimal class clusters that lead to better discriminant functions.

We propose a multilayer framework to obtain discriminant functions from nonlinear manifolds. An automated process (clustering) is used to obtain the pose-dependent classes for training. In the first layer, the projection surface is formed by using the entire data set. Clustering is performed on these data to form unsupervised training classes. These pose-dependent clusters are used for obtaining the discriminant features. The first layer gives a rough region of interest. The idea here is that this step divides the manifold into small regions of focus, where the manifold can now be assumed to be linear. In the second layer, we concentrate on this region of interest. Clustering approach is again used to obtain unsupervised smaller classes. These classes are now used to obtain a second set of discriminant features that can now be used to obtain finer pose estimation. By dividing the manifold into different layers and then to different classes, the smooth varying nonlinear manifold is now simplified into a locally linear problem.

Chutorian and Trivedi [1] give a detailed and organized survey about different head pose estimation methods and the evolution of this field. They give a description of the advantages and disadvantages of various methods. Pose estimation methods can be classified as appearance template methods, detector arrays, nonlinear regression methods, manifold embedding methods, flexible methods, geometric methods, tracking methods and hybrid methods. Appearance template methods [3, 4] compare the test image to a set of training images to find the closest template by using an image comparison method of choice. The number of training images can be extended at any time in appearance template methods. But, this method assumes that the region to be identified is localized and the localization error will decrease the accuracy of the method. Also, interpolation methods should be used for fine pose estimation [1]. A hierarchical graphical model for pose estimation from videos is proposed in [5]. This method provides a probability density function for a range of pose images. They are temporally modelled, and nonparametric density estimation is used for pose estimation. This method is successful in conditions like presence of occlusion, facial hair and glasses, blur and various facial expressions. Manifold embedding methods [6–8] follow low-dimensional manifolds modelled from continuous variations in head pose. New images can be embedded into this manifold, and then, template matching or statistical analysis is used for pose estimation.

Principal component analysis (PCA) is useful in the case when data lie on a linear subspace. If the data are not linear, PCA will not correctly extract the structure [6]. PCA is used for pose estimation by projecting the images into the PCA subspace [9]. Wavelet features were used here for better accuracy. Linear discriminant analysis (LDA) [10] maximizes the ratio of between-class variance to the within-class variance. In such cases, the class separability will be maximum. LDA can be used to obtain the discriminant features for pose estimation [11]. PCA can outperform LDA when the training data set is small and also PCA is less sensitive to different training data sets [12]. The kernalized translations of PCA and LDA are used for classification in [8] and in modelling multi-view faces [13]. For large data set problems, multiple LDA subspaces can be used for classification [14]. Here, the entire data set is divided into different equal sized sub-classes. LDA subspaces for each sub-class are then obtained. The unknown sample is projected onto all the subspaces, and classification is completed by a nearest neighbour search. They select the sub-classes in a random manner, and the overlapping data points between two adjacent classes are not considered. A multi-subspace analysis with discriminant analysis is used for classification in [15]. This paper discusses two types of multi-subspace analysis. The first method assumes that the images in the adjacent classes may be overlapped. A separate projection plane is used for the classification of these overlapped images. In the second method, images are projected into any one of the projection planes created by different predefined subspaces based on the neighbourhoods. One conclusion drawn here is that the way the sub-classes are formed affects the classification accuracy and the class formation in both layers are completely supervised. A two-stage head pose estimation framework is proposed in [8]. Here, the first stage estimates a rough region of interest based on appearance-based methods. In the second stage, geometric methods are used for fine pose estimation.

Nonlinear dimensionality reduction techniques—isometric feature mapping (Isomap) [16], locally linear embedding (LLE) [17] and Laplacian eigenmaps [18]—are applied for head pose estimation in [19–21]. The main disadvantage of these methods is the absence of a projection matrix to handle new test data points after the training process. Structural Laplacian eigenmaps [22] are used to learn the models representing a concept defined by a set of multivariate sequences. They use the intrinsic structure of the data to model the manifold. Biased manifold embedding (BME) [21, 23] framework for nonlinear dimensionality reduction uses the pose angle information for biasing neighbourhood calculation, resulting in better accuracy. But, this makes the process supervised since we need to use the pose information in the training stage. BME is possible only because completely annotated pose data are available for the particular database. In most real scenarios, we do not have the pose label information. It would be impractical for an end consumer to figure out pose information to train a system to one degree accuracy. So BME is infeasible in real-time scenarios. They adopted a generalized regression neural network (GRNN) with radial basis functions for nonlinear mapping. Synchronized submanifold embedding (SSE) [24, 25] based nonlinear dimensionality reduction method and random regression forests are used for pose estimation in [26]. SSE and random regression methods map the nonlinear data to linearly separable low-dimensional data. Interpolation techniques are used here to identify the missing range of pose values.

Gaussian process latent variable models are used for visualization of data in [27]. They model PCA as a gaussian process mapping from a latent space to a data space. A manifold-based two-layer (coarse and fine) pose estimation framework is used in [28]. The coarse pose estimation is based on supervised linear methods. First layer or coarse pose estimation is based on LDA, and it is used for broad classification of the test data into a rough region. The second-layer classification used linear regression function for fine pose estimation.

We propose a completely unsupervised multilayer framework for pose estimation. Clustering methods are used to form pose-dependent classes from the entire data set. Clustering algorithms separate data into different classes/subsets/categories with maximum within-class similarity and between-class separation [29, 30]. K-means clustering algorithm [31] is a well-known squared error type clustering, and in this method the cluster centroids are recomputed when a new sample joins a cluster. Clustered single-layer discriminant analysis is used for pose classification in [32]. Here, the clustering techniques are used for class formation for training. Automated class formation reduces the effort of manual class formation in a large database.

The unsupervised two-layer approach proposed in this paper helps to divide the manifold into small local regions. We used LDA for extracting the discriminant features and nearest neighbour algorithm for classification. LDA maximizes the ratio of between-class variance to the within-class variance in any particular data set, thereby guaranteeing maximal separability. LDA is proper for pattern classification if the number of training samples of each class is large [8, 12]. The main aim of LDA is to find a projection line which maximizes the ratio of between-class variance to the within-class variance. The between-class scatter matrix is defined as

$$S_{\mathrm{B}} = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{1}$$

and the within-class scatter matrix is defined as

$$S_{\mathrm{W}} = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T \tag{2}$$

where μ_i is mean image of class X_i and N_i is the number of samples in class X_i and c is the number of classes. If S_w is nonsingular, the optimal projection W_{opt} is chosen as the matrix with orthonormal columns which maximizes the ratio between the determinants of between-class scatter matrix to within-class scatter matrix of projected samples, where

$$W_{\mathrm{opt}} = \arg\max \frac{|W^T S_{\mathrm{B}} W|}{|W^T S_{\mathrm{W}} W|} = [w_1 w_2 \ldots w_m] \tag{3}$$

where $\{w_i | i = 1, 2, \ldots m\}$ is the set of generalized eigen vectors of S_{B} and S_{W} corresponding to the m largest eigen values $\{\lambda_i | i = 1, 2, \ldots m\}$, i.e.

$$S_{\mathrm{B}} w_i = \lambda_i S_{\mathrm{W}} w_i, \quad i = 1, 2, \ldots m \tag{4}$$

The maximum value of m is $c - 1$ [33].

Nearest neighbour algorithm with euclidean distance measurement is used for classification. This considers the euclidean distance between the feature vectors of training and testing samples. The sample in the training images with the shortest distance to the test sample will be the winner, and the test image is classified as the corresponding class of the training sample. The euclidean distance in n dimensional feature space between two points $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$ is defined by

$$d_e(a, b) = \sqrt{\sum_{i=1}^{n} (b_i - a_i)^2} \tag{5}$$

The decision boundaries of the decision regions produced by nearest neighbour technique are always piecewise linear because they consist of a number of line segments that are equidistant from a pair of samples of different classes [31].

2 Multi-subspace discriminant analysis

The main objective of the multi-subspace approach is to obtain locally linear portions of a nonlinear manifold by dividing data into specific groups. The entire manifold is used to find the first set of discriminant features to identify a rough match. Separate subspace(s) is/are used from this identified match neighbourhood for next level of feature extraction. In this second-layer subspace(s), the images would be well separated, and hence, classification should be simpler. Linear discriminant analysis (LDA) is used for feature extraction, and the nearest neighbour algorithm is used for classification in both layers. A conceptual drawing of the multi-subspace approach is shown in Fig. 1. This approach should help in classification of data points between adjacent classes by providing maximal separation between classes. Figure 2a shows the projection of two overlapping classes in the top two dimensions of the LDA subspace. The nonlinearity of the images can be observed from this projection, and the points inside the circle are overlapping images. Figure 2b shows the second-layer subspace projection of these images showing clear separation between the classes.

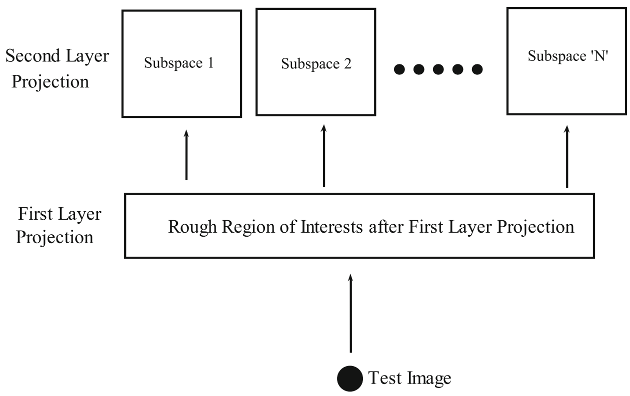

Fig. 1 Multi-subspace projection

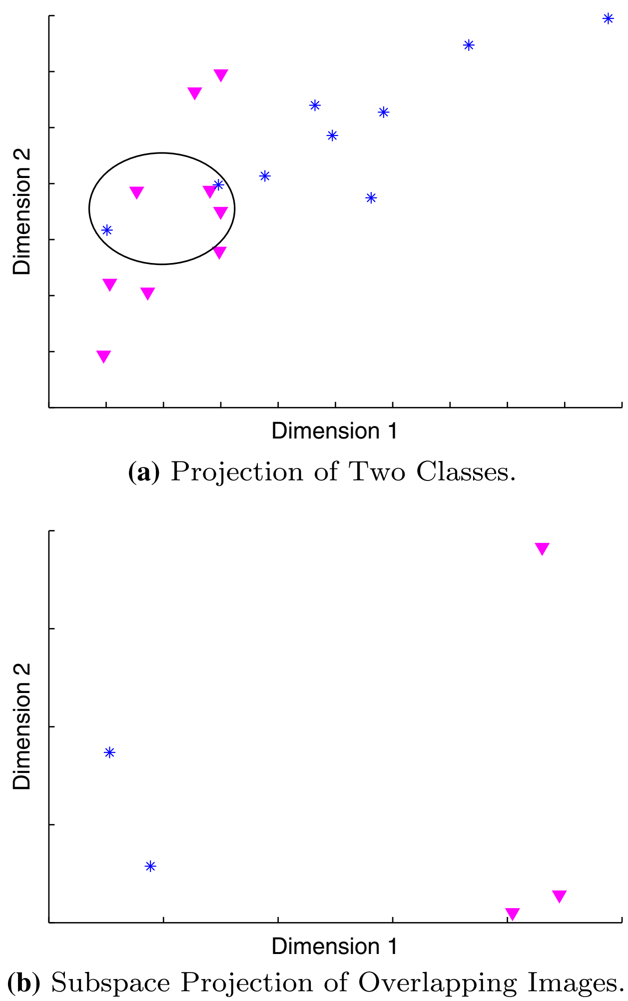

Fig. 2 Classification of data points in different projection planes. Separate projection plane results in more class separation for overlapping data points. **a** Projection of two classes. **b** Subspace projection of overlapping images

3 Clustered two-layer architecture

The accuracy/error rate of classification depends on the elements of the training database and the number of classes used for training and elements in each of those classes. This becomes critical when we try to find the discriminating features. Manual clustering does not explain or provide any logical or scientific base for forming classes from a smooth manifold. We propose clustering methods for automatic class formation for training. We use pose angles of a single person for clustering as shown in Fig. 3. The classes which now form should be based on the pose angles and hence will be pose dependent. To complete the class formation, we add the same pose angles of other persons in the training set to the corresponding classes as shown in Fig. 4. These classes are now used for training. The classes formed after clustering are used to obtain the discriminant functions. We project each image using these discriminant functions, and the nearest neighbour algorithm is used for classification. This approach can be considered as single-layer classification (projection to one common subspace).

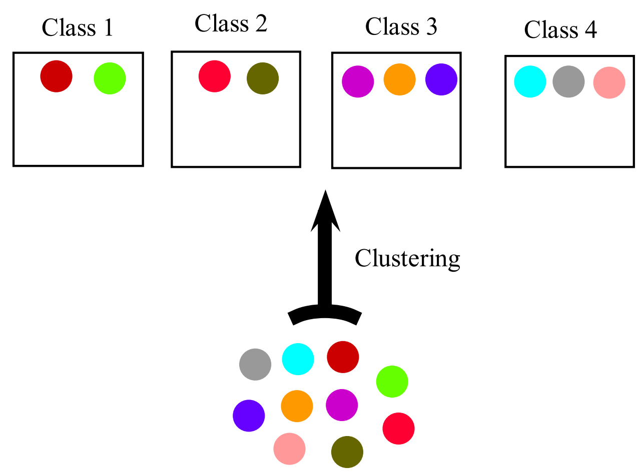

Fig. 3 Clustering—conceptual drawing (pose images of a single person is used for clustering)

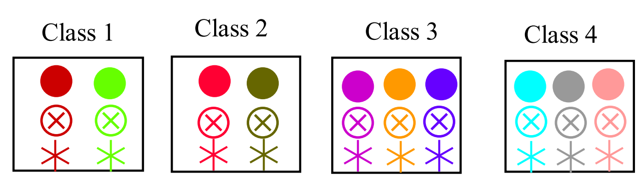

Fig. 4 Clustering—final class formation conceptual drawing (pose images of all other persons into corresponding classes). These final classes are used for training, and they will be purely pose dependent

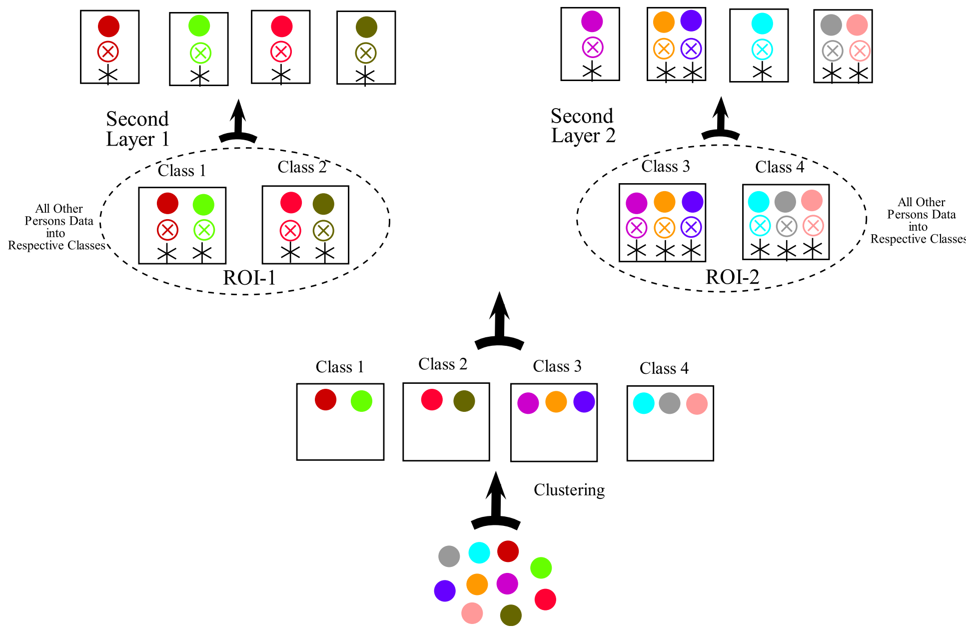

Fig. 5 Clustered two-layer architecture—conceptual drawing

Now we extend the pose estimation process to two layers. The first layer identifies a rough region of interest, and the second layer gives fine pose estimation. Classes formed after the first-layer clustering are used for finding the first set of discriminant functions. Classes close to each other are found based on the euclidean distance between the cluster centroids. Nearby classes are grouped together to form a second-layer region of interest which is again clustered into multiple classes. These classes are used for finding the second-layer discriminant functions. A conceptual drawing of this architecture is given in Fig. 5.

During testing, we first identify a rough region to which the test image could belong to. This rough area is shown as ROI-1 in Fig. 5. Second-layer 1 is considered for the next level classification because ROI-1 is in second-layer 1. This rough region of interest is now divided into different number of clusters as shown in Fig. 5.

4 Results and discussions

The proposed method is tested and evaluated on standard face pose image databases. CUbiC FacePix database [34, 35] and Pointing'04 [36] database are used for experimental analysis. Initially, we tested the approach in CUbiC FacePix database, and then, the Pointing'04 database is used for heterogeneous testing. CUbiC FacePix database consists of pose images of 30 different persons in different lighting conditions. Pose angle variation is from $-90°$ to $+90°$ per person. For testing, images captured from various pose angles with an ambient light source are used. This should ensure that the nonlinearity/variation is only due to change in pose. Leave-one-out testing strategy is followed here. One person's data are taken out from the database for testing, and the remaining data points are used for training. In each iteration, a different person's data are kept out for testing. We iterate so that each person's data are used for testing. The training image which is closest to the test image based on the euclidean distance measurement is the winner. We calculate the mean absolute error (MAE) for each iteration defined as

$$\mathrm{MAE} = \frac{1}{n}|\hat{\theta}_i - \theta_i| \tag{6}$$

where $\hat{\theta}_i$ is the estimated pose of the test input, θ_i is the original pose of the test image and n is the total number of samples used for testing.

To obtain the discriminant functions, the training data are divided into different classes. The easiest way to do this is to have each pose as its own class. But, this may not result in the best MAE. To test this, we manually create varying number of classes as shown in Table 1. To have benchmark results, the MAE of LDA projections using these classes is calculated and is as given in Table 2.

The clustering strategy is tested with K-means, fuzzy C-means [31] and hierarchical clustering [31] algorithms, the results for which are given in Tables 3, 4 and 5, respectively. It is clear that when the number of clusters increased, the error rate in pose estimation reduced. Clustering approach gives results comparable to manual clustering. Also, by using an automated clustering approach the effort of manual clustering in a database containing more than 5000 images is avoided.

Table 1 Manual clustering—cluster information

Total number of classes	Number of poses per persons per class	Cluster information
6	30	First class consists of pose angles from 1 to 30
		Second class consists of pose angles from 31 to 60 and so on
12	15	First class consists of pose angles from 1 to 15
		Second class consists of pose angles from 16 to 30 and so on
18	10	First class consists of pose angles from 1 to 10
		Second class consists of pose angles from 11 to 20 and so on
36	5	First class consists of pose angles from 1 to 5
		Second class consists of pose angles from 6 to 10 and so on
181	1	First class consists of pose angle 1
		Second class consists of pose angle 2 and so on

We consider the pose angle 0 only for the case where the total number of classes is 181. We neglect that angle in all other cases in both training and testing

Table 2 Mean absolute error for single-layer manual clustering using linear discriminant analysis

Number of classes	6	12	18	36	181
MAE (in degrees)	16.7378	6.7185	6.7943	5.5172	5.0236
Average time for classification (in seconds)	0.1782	0.7982	0.2108	0.2242	0.2774

Table 3 Single-layer classification using K-means clustering approach

Number of classes	6	12	18	36	75	150	181
MAE (in degrees)	13.6414	8.8029	7.5825	5.6006	5.2998	5.0538	5.0236
Average time for classification (in seconds)	0.2056	0.2067	0.2041	0.2135	0.2232	0.2390	0.2565

Table 4 Single-layer classification using fuzzy C-means clustering approach

Number of classes	6	12	18	36	75	150	181
MAE (in degrees)	14.5650	9.1168	6.9506	5.6856	5.5527	5.0700	5.0236
Average time for classification (in seconds)	0.2009	0.2051	0.2091	0.2134	0.2142	0.2334	0.2334

We evaluate the performance of clustering approach by varying the identity of the person under consideration for clustering. This is important since we could use training data of any of the persons under consideration. If accuracy were to depend on the person we choose for clustering, we would have a problem. Let us assume that the first person is to be tested. In this case, we could use any of the remaining persons data for clustering. Initially, we use the data of the second person for obtaining clusters. These classes are used for finding discriminant features, and MAE of pose estimation is calculated. This process is iterated such that we obtain MAE using clusters formed from each person in the database. Finding the average of these MAE values over 30 iterations should give us an idea about how person identity affects clustering process and hence the estimation accuracy. We create 150 classes using different clustering techniques, and the results of this single-layer classification are shown in Table 6. From Tables 3, 4, 5 and 6, it is clear that we can select pose angles of any person for clustering since the variation in MAE of pose estimation is low.

The nearest classes in the first layer are identified based on the euclidean distance between cluster centroids. These nearby classes are combined to form the second-layer region of interests (ROI). Data points in each ROI are clustered to get the second-layer classes. The number of ROIs and number of classes in each ROI are varied, and MAE of each combination is calculated. The best results obtained are illustrated in Table 7. The nonlinear manifold can be better represented by having more layers. Hence, having a third or fourth layer should increase the estimation accuracy. But for this, a very large data set is required.

Table 5 Single-layer classification using hierarchical clustering approach

Hierarchical clustering linkage criteria	Number of classes	6	12	18	36	75	150	181
Single	MAE (in degrees)	26.0094	15.9560	12.9153	9.1180	7.0770	5.1249	5.0236
	Average time for classification (in seconds)	0.2001	0.2049	0.2104	0.2121	0.2238	0.2430	0.2527
Average	MAE (in degrees)	14.7543	15.9560	7.7118	5.9720	5.2610	5.0357	5.0236
	Average Time for classification (in seconds)	0.2014	0.2049	0.2187	0.2137	0.2211	0.2393	0.2487
Complete	MAE (in degrees)	16.1263	9.8665	7.9483	5.8133	5.2794	5.0284	5.0236
	Average time for classification (in seconds)	0.2014	0.2055	0.2093	0.2135	0.2202	0.2401	0.2492
Median	MAE (in degrees)	14.6455	9.2302	7.8746	6.0483	5.2160	5.0169	5.0236
	Average time for classification (in seconds)	0.2014	0.2052	0.2091	0.2152	0.2196	0.2398	0.2484

Table 6 Single-layer classification using clustering approach by selecting pose angles of different persons for clustering—average MAE (in degrees)

Number of classes	150
K-means	5.0880
Fuzzy C-means	5.0681
Hierarchical	
Single	5.0411
Avereage	5.0409
Complete	5.0486
Median	5.0169

The proposed method is compared with existing face pose estimation approaches using grayscale features and is shown in Table 8. The clustered two-layer architecture approach performs better than all other methods except localized model using canonical correlation analysis (CCA), two-layer framework and supervised neighbourhood preserving embedding (NPE). In localized model using CCA [28], a priori neighbourhood information is given, and in supervised NPE, the out of sample extension cannot be performed. In two-layer framework [28], the first layer (coarse pose estimation) is performed using class-based supervised techniques and tested using overlapping regions. In contrast to this, the pose estimation in this proposed method is unsupervised.

4.1 Heterogeneous testing

In real-time cases, the experimental situations would be suddenly varying. The developers cannot predict all experimental situations in the beginning itself. So to ensure the robustness of the method, it is necessary to train the system with one database and should be tested using another database with entirely different experimental conditions. Based on these considerations, we used two different databases for heterogeneous testing. CUbiC FacePix database is used for training, and Pointing'04 database is used for testing. These two databases are different in their sizes and imaging conditions. The Pointing '04 database consists of different pose angles of 15 persons with variations of pitch and yaw angles in between −90 to +90 degrees. Pitch and yaw angles are varied as [−90, −60, −30, −15, 0, +15, +30, +60, +90] and [−90, −75, −60, −45, −30, −15, 0, +15, +30, +45, +60, +75, +90], respectively.

Table 7 Clustered two-layer architecture—MAE

Number of first-layer classes	Second layer		MAE (in degrees)	Average time for classification (in seconds)
	Number of regions of interests (ROI)	Number of classes in each ROIs		
181	25	2	5.7453	0.3637
		3	5.9134	0.3687
		4	5.5580	0.3747
150	25	2	5.9569	0.3563
		3	5.4011	0.3598
		4	4.6519	0.3545
75	25	2	6.1539	0.3478
		3	6.2279	0.3503
		4	4.5797	0.3399

Table 8 Comparison with existing methods

Sl.No	Method	MAE (in degrees)
1	Global model (CCA) [28]	10.03
2	Localized model (CCA) [28]	4.09
3	Two-layer framework [28]	4.14
4	Neighbourhood preserving embedding (NPE) [37]	8.20
5	Locality preserving projection (LPP) [37]	9.50
6	Supervised NPE [37]	4.40
7	Supervised LPP [37]	5.00
8	Multi-subspace discriminant analysis by considering overlapping images [15]	5.40
9	Multi-subspace neighbourhood discriminant classifier [15]	6.79
10	Manual class formation (Single layer)	5.02
11	Single-layer clustering strategy 1 [32]	13.42
12	Single-layer clustering strategy 2 [32]	5.02
13	Clustered two-layer architecture approach	4.5497

Table 9 Single-layer classification using K-means clustering approach—heterogeneous testing

Number of classes	6	12	18	36	75	150	181
MAE (in degrees)	44.3385	44.7231	41.9538	50.7385	50.3558	16.9538	14.4615
Average time for classification (in seconds)	0.2475	0.0077	0.182	0.1487	0.1928	0.2475	0.3127

Table 10 Clustered two-layer architecture—heterogeneous testing

Number of first-layer classes	Second layer		MAE (in degrees)	Average time for classification (in seconds)
	Number of region of interests (ROI)	Number of classes in each ROIs		
181	25	2	11.2308	0.8937
		3	12.4308	0.8777
		4	12.0108	0.8921
150	25	2	15.2923	0.9240
		3	14.5692	0.8438
		4	14.8000	0.8862
75	25	2	14.7435	0.8113
		3	12.7231	0.8335
		4	13.9692	0.8314

In CUbiC FacePix database, the images are varying with an interval of 1° in yaw direction. The pitch angle is not varying in this database. This larger database is used for training. The face pose images of Pointing'04 database with 0° pitch and yaw angles from -90° to $+90^\circ$ are used for testing. This subset of images of Pointing'04 database is aligned in the same manner as in the CUbiC FacePix database.

We evaluated the single-layer classification using K-Means clustering approach. The training classes are obtained from CUbiC FacePix database by K-means clustering approach. The subset images of Pointing'04 are used for testing. The evaluation results of this method are given in Table 9. We also evaluated the clustered two-layer approach using these two databases. The results of this method are given in Table 10. We compared this heterogeneous testing strategy with the two-layer framework proposed in [28], which uses the same heterogeneous testing strategy and databases. These comparison results are given in Table 11. Our single-layer clustering approach

Table 11 Comparison with existing method—heterogeneous testing

Sl.No	Method	Training database	Testing database	MAE (in degrees)
1	Two-layer framework [28]	CUbiC FacePix	Pointing'04	14.3
2	Single-layer clustering approach	CUbiC FacePix	Pointing'04	14.4615
3	Clustered two-layer approach	CUbiC FacePix	Pointing'04	11.2308

performance is very close to the two-layer framework proposed in [28], and a better result is obtained in clustered two-layer approach.

We also evaluated the processing times of all methods in a PC with Intel(R) Xeon(R) CPU of 3.70GHz and 8-GB RAM. We obtained the processing time of all pose estimations except heterogeneous two-layer approach as less than 0.5 s and this method consumes just less than 1 s.

5 Conclusion

In this paper, we proposed a method for classification of nonlinear face pose manifolds based on multiple subspaces using discriminant analysis. The main objectives of this work are to develop an unsupervised method to obtain the discriminant functions and to develop optimal class clusters for better discriminant functions. Clustering approaches were used to obtain the optimal class clusters. This automated process overcame the difficulty of manual clustering. The proposed multilayer framework was used to obtain discriminant functions. The first-layer classification identified a rough region of interest. In the second layer, the classification is concentrated on this region of interest. We performed experiments on standard face data bases to determine the effect of varying parameters on classification accuracy and to evaluate the processing time of our method. Heterogeneous testing was used to prove the robustness of the method and good results were obtained. The proposed approach was tested against other state of the art methods, which are supervised and comparable results were obtained.

References

1. Murphy-Chutorian E, Trivedi MM (2009) Head pose estimation in computer vision: a survey. IEEE Trans Pattern Anal Mach Intell 31(4):607–626
2. Czuprynski B, Strupczewski A (2014) High accuracy head pose tracking survey. In: Active media technology: 10th international conference, AMT 2014, Warsaw, Poland, August 11–14, 2014, Proceedings, vol 8610, Springer, Berlin, p 407
3. Niyogi S, Freeman WT (1996) Example-based head tracking. In: Proceedings of the second international conference on automatic face and gesture recognition, pp 374–378. IEEE
4. Beymer DJ (1994) Face recognition under varying pose. In: Proceedings CVPR'94., 1994 IEEE computer society conference on computer vision and pattern recognition, pp 756–761. IEEE
5. Demirkus M, Precup D, Clark JJ, Arbel T (2014) Probabilistic temporal head pose estimation using a hierarchical graphical model. In: Computer Vision–ECCV 2014, Springer, Berlin, pp 328–344
6. Cayton L (2005) Algorithms for manifold learning. University of California at San Diego Technical Report, pp 1–17
7. Sherrah J, Gong S, Ong E-J (2001) Face distributions in similarity space under varying head pose. Image Vis Comput 19(12):807–819
8. Junwen W, Trivedi MM (2008) A two-stage head pose estimation framework and evaluation. Pattern Recogn 41(3):1138–1158
9. McKenna SJ, Gong S (1998) Real-time face pose estimation. Real-Time Imaging 4(5):333–347
10. Duda RO, Hart PE, Stork DG (2012) Pattern classification. Wiley, New York
11. Kwak N, Choi S-I, Choi C-H (2008) Feature extraction for regression problems and an example application for pose estimation of a face. In: Campilho A, Kamel M (eds) Image analysis and recognition, Springer, Berlin, pp 435–444
12. Martínez AM, Kak AC (2001) Pca versus lda. IEEE Trans Pattern Anal Mach Intell 23(2):228–233
13. Chen L, Zhang L, Hu Y, Li M, Zhang H (2003) Head pose estimation using fisher manifold learning. In: AMFG, pp 203–207
14. Martina U, Roth PM, Horst B (2009) Efficient classification for large-scale problems by multiple lda subspaces. In: VISAPP (1), pp 299–306
15. Sankaran P, Hari CV (2014) Multi subspace analysis with supervised separable clusters for classification of smooth nonlinear manifolds. In: Eighth international conference on image and signal processing (ICISP 2014)
16. Tenenbaum JB, De Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323
17. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326
18. Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput 15(6):1373–1396
19. Raytchev B, Yoda I, Sakaue K (2004) Head pose estimation by nonlinear manifold learning. In: Proceedings of the 17th international conference on pattern recognition, ICPR 2004, vol 4, pp 462–466. IEEE
20. Hu N, Huang W, Ranganath S (2005) Head pose estimation by non-linear embedding and mapping. In: IEEE international conference on image processing, ICIP 2005, vol 2, pp II–342. IEEE
21. Balasubramanian VN, Krishna S, Panchanathan S (2008) Person-independent head pose estimation using biased manifold embedding. EURASIP J Adv Signal Process 2008:63
22. Lewandowski M, Makris D, Velastin SA, Nebel J-C (2014) Structural laplacian eigenmaps for modeling sets of multivariate sequences. IEEE Trans Cybern 44(6):936–949
23. Balasubramanian VN, Ye J, Panchanathan S (2007) Biased manifold embedding: a framework for person-independent head pose estimation. In: IEEE conference on computer vision and pattern recognition, CVPR'07, pp 1–7. IEEE
24. Yan S , Zhang Z, Fu Y, Hu Y, Tu J, Huang TS (2007) Synchronized submanifold embedding for person-independent precise 3D pose estimation. Paper presented at the 2007 Global Infotech Conference, Urbana-Champaign, Illinois, 6 Sept 2007

25. Yan S, Wang H, Yun F, Yan J, Tang X, Huang TS (2009) Synchronized submanifold embedding for person-independent pose estimation and beyond. IEEE Trans Image Process 18(1):202–210
26. Zhu Y, Xue Z, Li C (2014) Automatic head pose estimation with synchronized sub manifold embedding and random regression forests. Int J Signal Process Image Process Pattern Recogn 7(3):123–134
27. Lawrence ND (2004) Gaussian process latent variable models for visualisation of high dimensional data. Adv Neural Inf process Syst 16(3):329–336
28. Foytik J, Asari VK (2013) A two-layer framework for piecewise linear manifold-based head pose estimation. Int J Comput Vis 101(2):270–287
29. Rui X, Wunsch D et al (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16(3):645–678
30. Berkhin P (2006) A survey of clustering data mining techniques. In: Grouping multidimensional data, Springer, Berlin, pp 25–71
31. Gose E, Johnsonbaugh R, Jost S (1996) Pattern recognition and image analysis. Prentice-Hall Inc, Upper Saddle River
32. Hari CV, Sankaran P (2014) Face pose estimation for driver distraction monitoring by automatic clustered linear discriminant analysis. In: IEEE international conference on vehicular electronics and safety (ICVES), pp 100–105. IEEE
33. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720
34. Black JA Jr, Gargesha M, Kahol K, Kuchi P, Panchanathan S (2002) A framework for performance evaluation of face recognition algorithms. In: Proceedings of SPIE, vol 4862, p 164
35. Little D, Sreekar K, John B, Sethuraman P (2005) A methodology for evaluating robustness of face recognition algorithms with respect to variations in pose angle and illumination angle. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing
36. Nicolas G, Daniela H, Crowley JL (2004) Estimating face orientation from robust detection of salient facial features. In: ICPR International Workshop on Visual Observation of Deictic Gestures, Citeseer
37. BenAbdelkader C (2010) Robust head pose estimation using supervised manifold learning. In: Computer Vision–ECCV 2010, Springer, Berlin, pp 518–531