Efficient Graph Construction for Label Propagation based Multi-observation Face Recognition

F. Dornaika[1,2], A. Bosaghzadeh[1], and B. Raducanu[3]

[1] University of the Basque Country UPV/EHU, San Sebastian, Spain
[2] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain
[3] Computer Vision Center, 08193 Bellaterra, Barcelona, Spain

Abstract. Human-machine interaction is a hot topic nowadays in the communities of multimedia and computer vision. In this context, face recognition algorithms (used as primary cue for a person's identity assessment) work well under controlled conditions but degrade significantly when tested in real-world environments. Recently, graph-based label propagation for multi-observation face recognition was proposed. However, the associated graphs were constructed in an ad-hoc manner (e.g., using the KNN graph) that cannot adapt optimally to the data. In this paper, we propose a novel approach for efficient and adaptive graph construction that can be used for multi-observation face recognition as well as for other recognition problems. Experimental results performed on Honda video face database, show a distinct advantage of the proposed method over the standard graph construction methods.

1 Introduction

In the field of human-machine interaction, faces play a major role. For instance, socially oriented robots are specifically designed to support richer forms of interactions with humans. Their primary mission is to detect human presence, engage in an interaction and behave in a personalized manner. State-of-the-art face recognition techniques can achieve very high accuracy rates under controlled conditions. However, most of current face recognition systems lack robustness in uncontrolled environments (e.g., outdoor scenarios, homes, offices, etc.), since they are pretty sensitive to pose, lighting, occlusions and other variations (such as the presence of natural or artificial structures: beards, moustaches, glasses, etc.). Hence, the challenge is two-fold: to discriminate between different persons and at the same time to be able to recognize the same person affected by one or several of the aforementioned transformations. In particular, head pose problem has been one of the bottlenecks for most current face recognition techniques, because it changes significantly a person's appearance. In order to generalize the face recognition tools, it is mandatory to increase the robustness of the face recognition approach.

In Multimedia and Human Machine Interaction context, the system must continuously deal with an incoming flow of face images and has to guarantee a temporal coherence of a person's identity during the whole duration of the interaction process. In many cases, the difficulty arises from the fact that there is only a small time frame to capture a face with a high probability that the grabbed images do not contain the required frontal

face images. On the other hand, videos very often provide non-frontal faces. To this end two categories of approaches were proposed. The first category uses manifold learning paradigms [1, 2] in which the face subspace is constructed using many examples depicting subjects in different poses. The second category generates frontal face from the input image and then apply classic face recognition methods on the reconstructed frontal face image. The second category can be split into two main kinds of approaches: i) 3D morphable models [3], and ii) View-based methods [4]. View-based methods train a set of 2D models, each of which is designed to cope with shape or texture variation within a small range of viewpoints.

In [5], the authors propose a Local Linear regression method for pose invariant face recognition. The proposed method can generate the virtual frontal view from a given non-frontal face image. The whole non-frontal face image is partitioned into multiple local patches and then linear regression is applied to each patch for the prediction of its virtual frontal patch. The method requires the pose of the non-frontal pose as input in order to predict the frontal face. Following the approach of Active Appearance Models, [6] develops a face model and a rotation model which can be used to interpret facial features and synthesize realistic frontal face images when given a single novel face image. In [7], the authors address the non-frontal face recognition using morphable models.

As can be seen, 3D based and view based methods have many limitations. For instance, the 3D morphable models require 3D scans and have high computational load. The view-based methods very often require very tedious learning that require that the images are annotated by their face poses. In order to overcome the above limitations, multi-observation face recognition can offer an alternative [8–11]. In this case, the observations can be either a temporal sequence of face images (video sequence) or just a subset of images. Obviously, recognizing persons by using more than one face image can improve the performance of recognition systems since the test images contain more information that may include more views and more lighting variations that help reducing ambiguities that affect one single snapshot based recognition systems. Most of video-based face recognition methods use complicated training schemes in order to classify the multiple observations (e.g., [12]). In the context of semi-supervised learning, graph-based label propagation can be seen as a powerful tool that solves the multi-observation recognition problem. In [13], the authors proposed a graph-based label propagation method that can infer the labels of unknown observations by optimizing a penalty function based on label consistency. In [14], the authors extended the work of [13] by including the constraint that multiple observations have the same label. However, in both works the graph was constructed in a ad-hoc way, that is, it uses a KNN graph. In this paper, we propose a graph construction method that is based on efficient and adaptive coding scheme. We use the obtained graph in order to infer the label of multi-observation. The results are obtained with the public Honda face database.

The remainder of the paper is structured as follows: in Section 2, we provide some backgrounds about graph construction methods. Section 3 describes our proposed graph construction that is based on Weighted Regularized Least Square coding. Section 4 presents the recognition procedure. The experimental results on face recognition are presented in Section 5. Section 6 concludes the paper.

2 Background

The data graph is a powerful tool that encodes pairwise similarities among data samples. To this end, a weighted graph $G = (V; E; \mathbf{W})$ is constructed, where V denotes the set of N nodes of the graph corresponding to N data samples and $E \subseteq V \times V$ denotes the set of edges between nodes. For undirected graphs, $\mathbf{W}$ is a symmetric non-negative similarity matrix representing the weights of the edges, i.e., node i is connected to node j by an edge whose weight is equal to w_{ij} . An ideal similarity matrix, hence an ideal similarity graph G, is one in which nodes that correspond to points from the same subspace are connected to each other and there are no edges between nodes that correspond to points in different subspaces. Figure 1 illustrates a fully connected graph depicting the pairwise similarities among 6 face images.

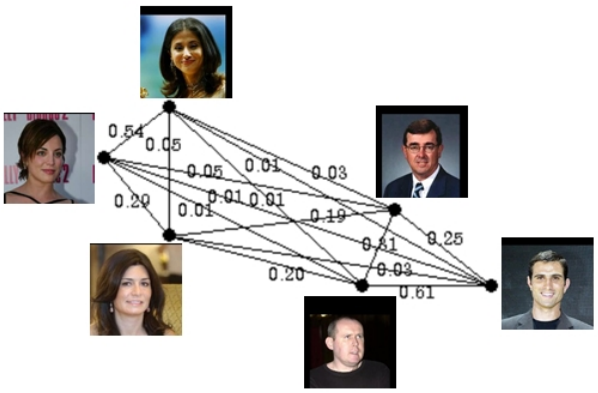

Fig. 1. A fully connected graph that encodes pairwise similarities among 6 face images.

2.1 KNN graph

The KNN graph is a well known scheme for constructing data graphs. Given N data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in \mathbb{R}^D$, one can build a nearest neighbor graph G to model the local geometrical structure. For each data point $\mathbf{x}_i$, we find its k nearest neighbors and put an edge between $\mathbf{x}_i$ and its neighbors. Let $N(\mathbf{x}_i) = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$ be the set of its k nearest neighbors. Thus, the weight matrix of G can be defined as follows:

$$W_{ij} = \begin{cases} sim(\mathbf{x}_i, \mathbf{x}_j) & \text{if } \mathbf{x}_j \in N(\mathbf{x}_i) \text{ or } \mathbf{x}_i \in N(\mathbf{x}_j) \\ 0, & otherwise \end{cases} \quad (1)$$

where $sim(\mathbf{x}_i, \mathbf{x}_k)$ is a real value that encodes the similarity between $\mathbf{x}_i$ and $\mathbf{x}_k$. Simple choices for this function are the Kernel heat and the cosine.

2.2 ϵ-ball graph

The ϵ-ball graph is similar to the KNN graph. However, instead of linking pairs according to the KNN rule, the neighbors of a given $\mathbf{x}_i$ are set to the data samples that belong to a sphere centered at $\mathbf{x}_i$ and having ϵ as radius.

2.3 L1 graph

In traditional graph construction process, the graph adjacency structure and the graph weights are derived separately (previous sections). In [15], the authors argue that the graph adjacency structure and the graph weights are interrelated and should not be separated. Thus it is desired to develop a procedure which can simultaneously completes these two tasks within one step. Instead of building a graph in two different processes of adjacency construction and graph weight calculation, [15] tried to unify them in one single process. This has the advantage that both relationship and degree of similarity can be simultaneously estimated. In their work, every sample is coded as a sparse linear combination of the rest of the training samples. If $\mathbf{y} \in \mathbb{R}^D$ denotes an input sample (test or training sample) and matrix $\mathbf{X} \in \mathbb{R}^{D,N}$ a database with N training samples, then the goal is to represent input $\mathbf{y}$ as a sparse linear combination of matrix X. Thus, it can be written as

$$\min ||\mathbf{a}||_1 \;\; s.t \;\; \mathbf{y} = \mathbf{X}\,\mathbf{a} \tag{2}$$

where $\mathbf{a} \in \mathbb{R}^N$ is the coefficient vector. Due to the presence of noise, Eq (2) will become:

$$\min ||\mathbf{a}||_1 \;\; s.t \;\; \|\mathbf{y} - \mathbf{X}\,\mathbf{a}\|_2 < \xi \tag{3}$$

which ξ represents a given tolerance error. After solving the above minimization problem, the components of the sparse vector $\mathbf{a}$ shows the contribution of each sample in reconstructing the input signal $\mathbf{y}$. The more similar a signal in the database to the sample, the bigger it's coefficient (absolute value). The samples in the database which are far from the input signal will have very small or zero coefficients. In this way the neighbors and their weights are calculated simultaneously. Therefore, by using the above coding for every training sample $\mathbf{x}_i$ with respect to the rest of the set, one can compute N sparse vector $\mathbf{a}_i$. From these coefficients, one can set the graph matrix $\mathbf{W}$ (w_{ij} is set to $|\mathbf{a}_i(j)|$).

There are different methods that can solve the above L_1 minimization problem like gradient projection [16], homotopy [17], iterative shrinkage-thresholding [18], proximal gradient, and alternating direction [19]. In [20] the authors reviewed those five representative L_1 minimization methods and compared them in speed and accuracy. The main focus was on the face recognition application. The conclusion shows that none of the above methods were able to outperform in all aspects of speed, accuracy and resistance of noise.

3 Proposed graph construction

As explained in the previous section there are many different methods to build a graph. Our objective is to provide an efficient tool for graph construction that has the same

advantages of L_1 graphs. In our proposed method, we construct the graph of a database by directly using the coding of any training image with respect to the rest of the set. We were inspired by recent advances in collaborative coding, namely the Weighted Regularized Least Square minimization method (WRLS) proposed in [21]. In this work, the authors proposed a linear coding scheme in order to classify samples according to the collaborative reconstruction error. Their proposed criterion is based on the sum of three parts: (i) L_2 norm of the reconstruction error, (ii) a regularization term set to the L_2 norm of the coefficients vector, (iii) a weighted sum of the squared coefficients. Since the weights are set to the distances between the test sample and the training samples, a kind of sparsity is included in the global criterion.

3.1 Weighted Regularized Least Square (WRLS) coding

In our work we use the following coding scheme in order to automatically generate the data graph:

$$\min_{\mathbf{a}} \left(\|\mathbf{y} - \mathbf{X}\,\mathbf{a}\|^2 + \sigma \sum_{i=1}^{N} p_i\, a_i^2 \right) \tag{4}$$

where σ is a regularization parameter having small positive value and the p_i's are non-negative weights. The above optimization problem has the following closed form solution:

$$\mathbf{a}^\star = (\mathbf{X}^T\,\mathbf{X} + \sigma\,\mathbf{P})^{-1}\mathbf{X}^T\,\mathbf{y} \tag{5}$$

where $\mathbf{P}$ is $N \times N$ diagonal matrix whose diagonal elements P_{ii} are set to p_i. In our work, we use the following formula for the weights:

$$P_{ii} = p_i = 1 - \exp(-\frac{\|\mathbf{y} - \mathbf{x}_i\|^2}{\gamma})$$

If the test sample $\mathbf{y}$ is far from the sample $\mathbf{x}_i$ then the weight of the unknown coefficient a_i tends to 1 so that the program in (4) attempt to get a small a_i. On the other hand, if the test sample is very close to the sample $\mathbf{x}_i$, the constraint on a_i is released.

3.2 WRLS graph

The detailed procedure for the WRLS graph construction is listed in Algorithm 1. Note that the constructed WRLS graph is a directed graph, i.e., the weight matrix $\mathbf{W}$ is asymmetric.

4 Multi-observation recognition based on label propagation

Label propagation is very often linked to the case of semi-supervised learning where the goal is to infer the unknown labels from the known ones using a given criterion [13]. Let C denotes the total number of classes. Let $\mathbf{X}_u$ (a $D \times r$ matrix) denote the r unknown observations. Let $\mathbf{X}_l$ (a $D \times N$ matrix) denote the N known observations (i.e., the training samples). The union of both data sets provides the data matrix $\mathbf{X} = (\mathbf{X}_l, \mathbf{X}_u)$.

Data: A given training sample set $\mathbf{X}$
Result: A weight matrix $\mathbf{W}$

Set the diagonal elements of $\mathbf{W}$ to zero ;
for $i = 1, \ldots, N$ **do**
  Pick the sample $\mathbf{x}_i$ from $\mathbf{X}$;
  $\mathbf{X}'$ = $\mathbf{X}$ - $\{\mathbf{x}_i\}$;
  Compute the $(N-1) \times (N-1)$ diagonal matrix $\mathbf{P}$;
  Calculate the $N-1$ vector $\mathbf{a}$ as $\mathbf{a} = (\mathbf{X}'^T \mathbf{X}' + \sigma \mathbf{P})^{-1} \mathbf{X}'^T \mathbf{x}_i$;
  for $j = 1, \ldots, N$ **do**
    if $i < j$ **then**
      Set $W_{ij} = |a_j|$
    else
      Set $W_{ij} = |a_{j-1}|$
    end
  end
end

Algorithm 1: WRLS graph construction.

The corresponding label matrix is denoted by $\mathbf{Y} = (\mathbf{Y}_l, \mathbf{Y}_u)$ (a $C \times (N+r)$ matrix). Each column vector $\mathbf{y}_i$ of $\mathbf{y}$ is a vector characterizing the probabilities of the sample $\mathbf{x}_i$ belonging to different classes, namely,

$$y_i(c) = p(c|\mathbf{x}_i); c = 1, 2, \ldots, C$$

where $p(c|\mathbf{x}_i)$ is the posterior probability of the class c for the given sample $\mathbf{x}_i$. For a labeled sample $\mathbf{x}_i$, $y_i(c) = 1$ if $\mathbf{x}_i$ belongs to the c^{th} class; $\mathbf{y}_i(c) = 0$, otherwise.

The problem of label propagation is to infer the label matrix $\mathbf{Y}_u$ given the whole data $\mathbf{X} = (\mathbf{X}_l, \mathbf{X}_u)$ and the known label matrix $\mathbf{Y}_l$. This can be achieved by minimizing the following criterion:

$$\min E(\mathbf{Y}) = \sum_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij} \tag{6}$$

An explanation of this objective is as follows. When the samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar, namely, the graph weight W_{ij} is large, the distance between $\mathbf{y}_i$ and $\mathbf{y}_j$ should be small in order to minimize the objective, namely the class information of the sample $\mathbf{x}_i$ and $\mathbf{x}_j$ should be similar.

The objective can be further rewritten as

$$\begin{aligned} \min E(\mathbf{Y}) &= \sum_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij} && (7) \\ &= trace(\mathbf{Y}\,\mathbf{D}_{row}\,\mathbf{Y}^T + \mathbf{Y}\,\mathbf{D}_{col}\,\mathbf{Y}^T - \mathbf{Y}\,\mathbf{W}\,\mathbf{Y}^T - \mathbf{Y}\,\mathbf{W}\,\mathbf{Y}^T) && (8) \\ &= trace(\mathbf{Y}\,\mathbf{L}\,\mathbf{Y}^T) && (9) \end{aligned}$$

where $\mathbf{D}_{row}$ is a diagonal matrix whose diagonal elements are the row sums of the corresponding rows of $\mathbf{W}$, and $\mathbf{D}_{col}$ is a diagonal matrix whose diagonal elements are

the column sums of the corresponding columns of $\mathbf{W}$. $\mathbf{D}_{row} - \mathbf{W}$ and $\mathbf{D}_{col} - \mathbf{W}^T$ are the row and column Graph Laplacian matrices respectively. It is obvious that the matrix $\mathbf{L} = \mathbf{D}_{row} + \mathbf{D}_{col} - (\mathbf{W} + \mathbf{W}^T)$ is symmetric.

Since the r observations have the same unknown label, the unknown label matrix $\mathbf{Y}_u$ will have C configurations $(\mathbf{Y}_u(1), \ldots, \mathbf{Y}_u(C)$ where $\mathbf{Y}_u(c)$ has only the c^{th} row equal to ones and the the rest of the rows are zeros. Therefore, the whole label matrix $\mathbf{Y} = (\mathbf{Y}_l, \mathbf{Y}_u)$ can be written as $\mathbf{Y} = (\mathbf{Y}_l, \mathbf{Y}_u(c))$ where $\mathbf{Y}_l$ is constant. To infer the label of the unknown observations $\mathbf{X}_u$, the following formula can be used:

$$c^\star = \arg\min E(\mathbf{Y}_c) \tag{10}$$

where $\mathbf{Y}(c) = (\mathbf{Y}_l, \mathbf{Y}_u(c))$. Thus, the optimal label is inferred using C evaluations of the term $E(\mathbf{Y}_c)$. The procedure for the multi-observation recognition based on the WRLS graph is illustrated in Algorithm 2.

Data: A set of multiple observations $\mathbf{X}_u$, a training set $\mathbf{X}_l$ and their labels $\mathbf{Y}_l$
Result: The label of the unknown observations c*

Compute the WRLS graph over the data $\mathbf{X} = (\mathbf{X}_l, \mathbf{X}_u)$ (Algo 1) ;
Estimate the label $c^\star$ using Eq. (10)

Algorithm 2: Multi-observation recognition via WRLS graph based label propagation.

5 Experimental Results

Data preparation Our approach has been tested on the Honda video database (HVDB) [22, 23]. HVDB has been acquired for the purpose of face tracking and recognition. It depicts persons sitting in front of a camera in a totally uncontrolled environment and performing unconstrained in-plane and out-of-plane head motion. Some samples are depicted in figure 2. The resolution of the images is 640x480 pixels and the videos were recorded at 15 frames per second. We selected from this database a subset of 22 video clips belonging to 22 different persons.

The used dataset contains 2317 images organized in 22 classes, with an average of 100 images per class. The cropped images were resized to 50×50 pixels. The cropped images are normalized using the zero-mean unit-variance. A sample gallery is depicted in figure 3.

We conducted two groups of experiments. In the first group, we use single snapshots in order to recognize faces using a manifold representation followed by the Nearest Neighbor classifier. In the second group of experiments we use the label propagation theory in order to recognize faces based on several observations. In the latter group, we consider two cases: the KNN graph and the proposed method for automatic graph building.

Fig. 2. Some samples from the Honda Video database.

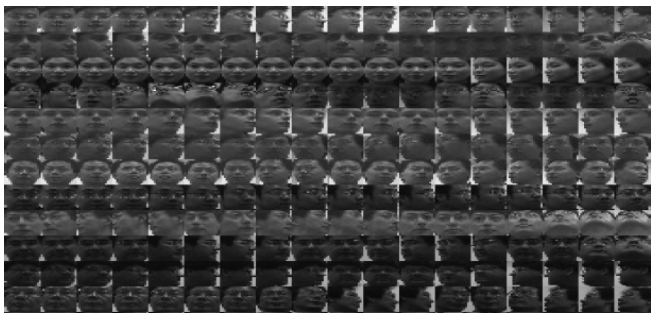

Fig. 3. Some samples of cropped face images from the Honda Video database.

Single image face recognition. For single snapshot face recognition evaluation, we applied a manifold representation approach. This implies that the original, high-dimensional data, are embedded into a low-dimensional subspace, without any relevant loss of information. The projection function could be either linear or non-linear. More concrete, we used the following manifold learning techniques: Principal Component Analysis (PCA), Locally Linear Embedding (LLE) [24], Locality Preserving Projections (LPP) [25] and Laplacian Eigenmaps (LE) [26]. PCA and LPP belong to the category of linear embedding techniques, while LLE and LE are non-linear. The PCA method estimates orthogonal projection directions that maximize the variance of original data. The LPP method searches a linear projection that preserve the locality of the neighboring data. The LLE algorithm seeks the nonlinear embedding in a neighborhood preserving manner by exploiting the local symmetries of linear reconstructions, and seeking the optimal weights for local reconstruction. The embedding is obtained using the estimated local weights. The LE method seeks the nonlinear embedding by preserving locality. It should be noticed that the LPP method is the linearized version of LE.

For classification purposes, we adopted 10 random splits of data. We split the data in several ratios for training and test: $10\% - 90\%$, $20\% - 80\%$, $30\% - 70\%$, $40\% - 60\%$ and $50\% - 50\%$. The reason we decided to start with such a low percentage of training images (10%) is motivated by the fact that we have a pretty high number of instances per class. The classification in the embedded space has been carried out using the Nearest Neighbor (NN) approach. The recognition rates for the case of single snapshot recognition are reported in table 1. We can observe that (i) the PCA technique has provided the best recognition results for both the cropped and rectified faces; (ii) the LLE method has provided the worst results. This is very consistent with the fact that LLE is not very suited for classification tasks.

Multi-observation image face recognition. The experimental setup is similar to the one used in single image face scenario. We again use the ten random splits and report the average recognition rate over the ten random splits. We adopted three sizes for the number of test images of the same person: $r = 3$, 4 and 5 images. For every training percentage and for every value of r, several hundreds of subsets are picked at random from the test set and recognized by the algorithm provided in Section 4. Tables 2,3, and 4 summarize the performance for the training/test ratios $10\% - 90\%$, $20\% - 80\%$, and $30\% - 70\%$, respectively. In each of these tables, we compare two methods: (1) the KNN graph based label propagation and (2) our proposed graph-based label propagation (Section 4). For the KNN graph based method, several values for K were tested and the best performance were reported in these Tables. As can be seen the use of our proposed graph construction method has improved the recognition performance. Furthermore, this improvement is more obvious when the training size and/or the size of the multiple observation becomes very small.

Table 5 summarizes the performance of the single image face recognition scheme (PCA) and the Multi-observation face recognition scheme (proposed graph-based label propagation). For the latter scheme, the number of test images is kept fixed to 3 ($r = 3$).

Algorithms' complexity. We have measured the computing time of the proposed graph construction method. To this end, we used the training percentage of 10% together with $r = 3$ images. In other words, the graph size is 234 images. We used a non-optimized MATLAB code running on a PC equipped with an Intel Core i7 CPU at 2.93 Ghz.

Table 6 summarizes the performance of three methods for graph construction: (i) the KNN graph, (ii) the L_1 graph, and (iii) our proposed method. The first column depicts the CPU time associated with the graph construction stage, the second column depicts the CPU time associated with the recognition step (label propagation), and the third column depicts the recognition rate. Although the L_1 provided more accurate recognition rate, our proposed graph construction method was 43.3 times faster. This makes the online face recognition based on L_1 graph method unfeasible whereas this is still feasible with our proposed graph construction.

Train	10%	20%	30%	40%	50%
PCA	**74.73%**	**85.86%**	**91.67%**	**94.24%**	**95.85%**
LLE	34.18%	42.71%	47.84%	53.99%	64.53%
LPP	67.54%	80.40%	86.36%	89.62%	91.64%
LE	68.82%	77.48%	82.34%	85.52%	87.62%

Table 1. Best average recognition accuracy using manifold representation and the NN classifier.

Number of observations (Images)	3	4	5
KNN graph	80.92	85.37	88.21
Proposed graph	**84.28**	**87.06**	**88.81**

Table 2. Method comparison for multi-observation recognition. The size of training is 10%.

Number of observations (Images)	3	4	5
KNN graph	92.53	95.64	96.67
Proposed graph	**95.52**	**97.33**	**97.98**

Table 3. Method comparison for multi-observation recognition. The size of training is 20%.

6 Conclusions

In this paper, we proposed a novel approach for efficient and adaptive graph construction that can be used for multi-observation face recognition as well as for other recognition problems. Experimental results performed on Honda video face database, show a

Number of observations (Images)	3	4	5
KNN graph	96.21	97.95	98.65
Proposed graph	**97.75**	**98.74**	**99.67**

Table 4. Method comparison for multi-observation recognition. The size of training is 30%.

Training data set	10%	20%	30%
Single snapshot (PCA space)	74.73	85.86	91.67
Proposed scheme	**84.28**	**95.52**	**97.75**

Table 5. Method comparison. Second row: Recognition rates for singe snapshot recognition (PCA followed by NN classifier). Third row: Proposed method for multi-observation recognition using three face images. For the latter scheme, the number of test images is kept fixed to 3 ($r = 3$).

distinct advantage of the proposed method over the standard graph construction methods.

References

1. Cai, D., He, X., Zhou, K., Han, J., Bao, H.: Locality sensitive discriminant analysis. In: International Joint Conference on Artificial Intelligence. (2007)
2. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500) (2000) 2319–2323
3. Blanz, V., Vetter, T.: Face recognition based on fitting a 3D morphable model. IEEE Transactions on PAMI (2003)
4. Cootes, T.F., Wheeler, G.V., Walker, K.N., Taylor, C.J.: View-based active appearance models. Image and Vision Computing (2002)
5. Chai, X., Shan, S., Chen, X., Gao, W.: Locally linear regression for pose-invariant face recognition. IEEE Trans. on Image Processing **16**(7) (2007)
6. Shan, T., Lovell, B.C., Chen, S.: Face recognition robust to head pose from one sample image. In: IEEE International Conference on Face and Gesture Recognition. (2006)
7. Blanz, V., Grother, P., Phillips, P.J., Vetter, T.: Face recognition based on frontal views generated from non-frontal images. In: IEEE International Conference on Computer Vision and Pattern Recognition. (2004)
8. Lee, K., Ho, J., Yang, M.H., Kriegman, D.: Video-based face recognition using probabilistic appearance manifolds. In: CVPR. (2003) 313–320
9. Zhou, S., Krueger, V., Chellappa, R.: Probabilistic recognition of human faces from video. Computer Vision and Image Understanding **91**(1-2) (2003) 214–245
10. Hewitt, R., Belongie, S.: Active learning in face recognition: Using tracking to build a face model. In: V4HCI, New York City (2006)
11. Manyam, O.K., Kumar, N., Belhumeur, P., Kriegman, D.: Two faces are better than one: Face recognition in group photographs. In: International Joint Conference on Biometrics (IJCB), Washington, DC (October 2011)
12. Liu, X., Chen, T.: Video-based face recognition using adaptive hidden markov models. (2003) 340–345
13. D. Zhou, O. Bousquet, T.N.L.J.W., Schlkopf, B.: Adv. neural inf. process. syst. In: Learning with local and global consistency. (2004)

	Graph construction (CPU time)	Recognition (CPU time)	Recognition rate
KNN graph	0.0035 (seconds)	0.004 (seconds)	80.92 %
L_1 graph	24.06 (seconds)	0.004 (seconds)	85.46 %
Proposed WRLS graph	0.498 (seconds)	0.004 (seconds)	84.43 %

Table 6. CPU time associated with three graph construction methods. The size of the graph is 234 images. The number of unlabeled images is 3.

14. Kokiopoulou, E., Frossard, P.: Graph-based classification of multiple observation sets. Pattern Recognition **43**(12) (2010) 3988–3997
15. Yan, S., Wang, H.: Semi-supervised learning by sparse representation. In: SIAM Conference on Data Mining. (2009) 792–801
16. Figueiredo, M.A.T., Nowak, R., Wright, S.: Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. Selected Topics in Signal Processing, IEEE Journal of **1**(4) (2007) 586–597
17. Malioutov, D., Cetin, M., Willsky, A.: Homotopy continuation for sparse signal representation. In: Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on. Volume 5. (2005) v/733–v/736 Vol. 5
18. Combettes, P., Wajs, V.: Signal recovery by proximal forward-backward splitting. Multiscale Modeling and Simulation **4**(4) (2005) 1168–1200
19. Yang, J., Zhang, Y.: Alternating direction algorithms for l1-problems in compressive sensing. Technical report (2009)
20. Yang, A.Y., Sastry, S.S., Ganesh, A., Ma, Y.: Fast l1-minimization algorithms and an application in robust face recognition: a review. Technical report (2010)
21. Waqas, J., Yi, Z., Zhang, L.: Collaborative neighbor representation based classification using l2-minimization approach. Pattern Recognition Letters **34** (2013) 201–208
22. Lee, K.C., Ho, J., Yang, M.H., Kriegman, D.: Visual tracking and recognition using probabilistic appearance manifolds. Computer Vision and Image Understanding **99** (2005) 303–331
23. Lee, K.C., Kriegman, D.: Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In: IEEE International Conference on Computer Vision and Pattern Recognition. (2005)
24. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500) (2000) 2323–2326
25. He, X., Niyogi, P.: Locality preserving projections. In: Conference on Advances in Neural Information Processing Systems. (2003)
26. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation **15**(6) (2003) 1373–1396