Face Clustering in videos based on Spectral Clustering Techniques

Christina Chrysouli, Nicholas Vretos and Ioannis Pitas
Department of Informatics
University of Thessaloniki
Thessaloniki, Greece 54124
Email: pitas@aiia.csd.auth.gr

Abstract—**In this paper we propose a novel algorithm for face clustering using spectral graph clustering in order to split and merge a similarity graph. The proposed method makes use of the mutual information-based image similarity. Face clusters are formed based on spectral graph clustering in a two step process. We begin by partitioning the dataset into clusters. A novel adaptive way is proposed for spectral clustering. Then merge is performed using spectral graph clustering on the partitioned clusters, by considering merging only two clusters at a time. Experiments on various video databases containing actors' facial images are conducted. The evaluation of the face clustering provided very good results.**

I. INTRODUCTION

The aim of face clustering is to partition a large set of facial images into clusters, such that facial images of the same person (to be called actor) are placed in the same cluster, while facial images of different actors are placed into different ones. Face clustering is an important application on extracting semantic information in videos and can be used in a wide variety of video analysis applications, such as video indexing [1], automatic cast listing in movies [2] and preprocessing for face recognition [3].

So far, some interesting algorithms have been proposed for face clustering, but most of them make use of calibrated images, like the one presented in [4]. In [5] the authors developed a distance metric for face clustering algorithms, which is invariant to affine transformations. In [6], a method for face clustering was proposed using the results of a face detector/tracker and, at the same time, some heuristics based on the actors' appearance. Another approach was followed in [3], focusing on automatic detection and clustering of human faces, based on principal component analysis (PCA).

Spectral graph clustering [7], refers to a class of graph techniques, which rely on the eigenanalysis of the adjacency matrix or the Laplacian matrix of a similarity graph, in order to partition graph nodes (facial images in our case) in disjoint clusters, with nodes belonging to the same cluster having high similarity and nodes from different clusters having low similarity. Spectral graph clustering has been widely used in many image clustering applications [8].

Our approach uses the normalised cut criterion [9], in order to partition a set of facial images into clusters. The initial dataset is divided recursively into subsets (clusters), until a given cluster homogeneity, which is defined according to a suitably chosen similarity threshold, is attained. Furthermore, the method examines if merging some of the produced clusters is beneficial, by using spectral graph clustering on already formed clusters.

The rest of this paper is organised as follows. Section 2 presents the proposed clustering algorithm in detail and also an adaptive threshold choice is exploited. In Section 3, experimental results of the face clustering algorithm are described. Finally, in Section 4, conclusions are drawn and future work is discussed.

II. SPLIT-MERGE FACIAL IMAGE CLUSTERING

Mutual information between two probability distributions, namely X and Y, can be defined as the information shared between these distributions:

$$I(X;Y) = H(X) + H(Y) - H(X,Y), \tag{1}$$

where $H(X)$, $H(Y)$ are the entropy of X and Y respectively, and $H(X,Y)$ is the joint entropy:

$$H(X) = -\sum p(x)\log p(x), \tag{2}$$

$$H(X,Y) = -\sum p(x,y)\log p(x,y), \tag{3}$$

where $p(x)$ is the marginal probability of X and $p(x,y)$ is the joint probability of X and Y.

Mutual information, measures how much information can be obtained about one random variable by observing another.

In this paper, normalised mutual information (NMI) [10] is used as a measure of image similarity:

$$NMI(X;Y) = \frac{H(X)+H(Y)}{H(X,Y)}. \tag{4}$$

A similarity $N \times N$ matrix is created from N input facial images forming the facial image set V, in the same way as in [6], using (4).

Considering the facial images as nodes in an undirected weighted graph, we use the normalised cut criterion [9], in order to partition the facial images into clusters. The weight $W_{i,j}$ on each graph edge connecting any two nodes i and j

is defined to be their NMI value. The normalised cut criterion measures both the total dissimilarity among different image subsets, as well as the total similarity within the same subsets.

This criterion can be calculated by solving the generalised eigenvalue problem [9]. Let $G(V, E)$ be the above mentioned undirected weighted graph, whose nodes are partitioned into subsets $A \subseteq V$, $B = V - A$, by removing the edges connecting the nodes in A with the ones in B. Then, the total weight of the edges that have been removed gives the dissimilarity between the two sets (i.e. the cut):

$$cut(A, B) = \sum_{i \in A,\, j \in B} W_{i,j}. \tag{5}$$

The above criterion, though, is prone to cutting out small groups of isolated nodes. In this paper we have used the normalised cut criterion instead, defined as:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}, \tag{6}$$

where $assoc(A, V)$ is the total weight between edges in A component and all nodes in the graph:

$$assoc(A, V) = \sum_{i \in A, v \in V} W_{i,v}. \tag{7}$$

$assoc(B, V)$ is computed similarly. In order to minimise the normalised cut, we solve the generalised eigenvalue problem [11]. Let $\mathbf{D}$ be a diagonal $N \times N$ matrix having the sum $d_{ii} = \sum_j W_{i,j}$ on its main diagonal. Then, the generalised eigenvalue problem is defined as:

$$(\mathbf{D} - \mathbf{W})\mathbf{v} = \lambda \mathbf{D} \mathbf{v}. \tag{8}$$

This generalised eigenvalue problem, has a trivial solution for $\mathbf{v} = \mathbb{1}$ and $\lambda = 0$. We generally omit this solution and we take the second smallest eigenvalue, also known as algebraic graph connectivity [12]. It can be proven that the entries $v(i)$, $i = 1, ..., N$ of the eigenvector V associated with the second eigenvalue [11], define a function, whose domain is the nodes of the graph $G(V, E)$ and is related to the similarity structure, as defined by matrix $\mathbf{W}$. Thus, the eigenvector with the second smallest eigenvalue is used to split the set V of facial images in two clusters A, B. More specifically, images that have eigenvalues above the mean value of the eigenvector are placed together in a cluster, while images that have equal or lower eigenvalues are placed in another cluster. The criterion that is used in the proposed method, in order to decide if further partition in the dataset is needed, is the median value of the similarity matrix $\mathbf{W}_A$, $\mathbf{W}_B$ of each subset A, B. $\mathbf{W}_A$, $\mathbf{W}_B$ are formed from $\mathbf{W}$, by deleting the rows and columns corresponding to the nodes not in the subset A, B, respectively, provided by the normalised cut algorithm. Moreover, we calculate the median value of a matrix as follows. We first compute the median value of each row m_i, corresponding to the NMI value of an image to all others, and then compute the median of all m_i's:

$$med\left(\{med(\mathbf{W}_i) \,|\, i = 1, ..., N\}\right), \tag{9}$$

where $\mathbf{W}_i$ is the i-th column of matrix $\mathbf{W}_A$ or $\mathbf{W}_B$. If the median of $\mathbf{W}_A$ is lower than a threshold T, then further partition is needed, since the subset A is inhomogeneous. Same happens for subset B. The threshold is adapted in every recursion of the algorithm as will be described later on. Threshold T may vary in the various split iteration.

Summarising, the proposed split algorithm for partitioning a set of facial images in homogeneous clusters consists of the following steps:

1) Given a set of facial images, compute $\mathbf{W}$ and $\mathbf{D}$ matrices as described earlier.
2) Solve the generalised eigenvalue problem: $(\mathbf{D} - \mathbf{W})\mathbf{v} = \lambda \mathbf{D}\mathbf{v}$.
3) Use the eigenvector with the second smallest eigenvalue in order to split the set of images into two subsets.
4) Decide if the current subsets need to be repartitioned, by checking whether the median value of the produced similarity submatrices is lower the current threshold T_i.
5) Adapt the threshold T_i to T_{i+1}.
6) Go to step 1 with $\mathbf{W} = \mathbf{W}_A$, $\mathbf{D} = \mathbf{D}_A$ and $\mathbf{W} = \mathbf{W}_B$, $\mathbf{D} = \mathbf{D}_B$.

The aim of this algorithm is to partition the facial image set in such way, so that all images of the same person are placed together in the same cluster, while others are placed in different clusters.

In [6], a fixed threshold $T_i = T$ was used, in order to decide if subsequent partition was required. We have noticed that the clustering performance was extremely sensitive to fixed threshold changes. Very small threshold changes led to different number of clusters.

In our experiments, better clustering results were achieved when the threshold was reduced exponentially in every split recursion. In Figure 1, an example of how we partition a set of images is presented. The number inside each box represents the number of facial images in each recursion step of the split algorithm, while the numbers next to edges represent the threshold used in order to decide if further partition is required. The threshold is reduced exponentially at each recursion step:

$$T_i = T_{i-1} e^{-\lambda}, \tag{10}$$

where i is the level of the tree and λ is a small positive number. The basic intuition behind the threshold adaptation is that, as cluster cardinality gets smaller after each bipartition, it is reasonable to expect that the similarity between facial images in the same subset should increase. Thus, in order to end the partition before the algorithm splits the set into many unnecessary facial image clusters, we reduce the threshold.

Our experiments showed that, often, facial images that belong to different persons mix together, in the same cluster. In order to avoid that, we can overpartition the initial set into more clusters than persons, by suitably choosing the threshold value T_i in the previous algorithm. Although, this approach may result in oversplitting the facial image set, that belongs to the same person into more than one clusters, it is more likely that each cluster will consist of facial images from solely one

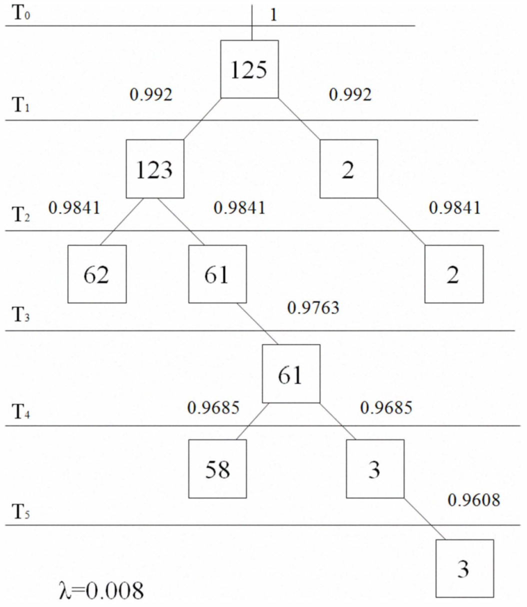

Fig. 1. Recursive partition tree.

person. Once the set is partitioned using the aforementioned method, an algorithm that merges the resulting clusters can be applied, aiming to merge clusters that, although consist of facial images of the same person, have been placed into separated clusters in the cluster split step. In order to merge the facial image clusters, we use spectral graph clustering, as we did previously for partitioning the set of facial images. More specifically, we decide if each facial image cluster should be merged into any other cluster, by computing the $\mathbf{W}$ and $\mathbf{D}$ matrices formed for each pair of facial image clusters and then solve the generalised eigenvalue problem using the similarity data of only the facial images belonging to those two clusters.

Suppose that we have two clusters A and B containing facial images. If, after the spectral analysis, the new cluster A' and/or B' contains facial images from both A and B clusters, then we merge the initial clusters. If both clusters A', B' continue to have the same facial images as A, B respectively, another criterion is applied to decide if they should be merged or not. The criterion is based on eigenvector entries $v(i)$, $i = 1, ..., N$. As we already know, we can assign to each graph node the corresponding entry $v(i)$ of the eigenvector $\mathbf{v}$.

To decide whether the facial image clusters should be merged, the square difference of the two means should be less than a threshold T:

$$\left(\frac{1}{|A|}\sum_{i\in A} v(i) - \frac{1}{|B|}\sum_{j\in B} v(j)\right)^2, \qquad (11)$$

where A, B the two partitions known a priory from the split process, and $|\cdot|$ denotes set cardinality. Moreover, as facial image clusters do not typically have the same cardinality, we normalise this difference, by dividing (11) with facial image clusters A, B cardinalities:

$$\alpha = \frac{1}{|A|+|B|}\left(\frac{1}{|A|}\sum_{i\in A} v(i) - \frac{1}{|B|}\sum_{j\in B} v(j)\right)^2 < T. \qquad (12)$$

As it can easily be seen, sometimes, a facial image cluster can be merged with more than one other clusters. In that case, we arbitrarily choose to merge cluster i with the cluster that have the smallest j value.

In other words, if facial images in two different clusters are well separated, then no merge is needed. On the other hand, if images are not well separated or change clusters, then merge is required. We can now summarise the steps of the merging process:

1) Given a partitioned set of facial images into clusters, compute $\mathbf{W}$ and $\mathbf{D}$ matrices for each pair of clusters.
2) Solve the generalised eigenvalue problem: $(\mathbf{D} - \mathbf{W})\mathbf{v} = \lambda\mathbf{D}\mathbf{v}$.
3) Decide if two subsets need to be merged together, by computing α from (12).
4) Merge the clusters if after the spectral analysis, the new clusters A' and/or B' contain facial images from both the initial A and B clusters, or in the opposite case, if the value of α is lower than a given threshold T.

The purpose of merging is to place together images of the same person, that may have been separated during cluster split.

III. Experiments

In order to test the performance of the proposed facial image clustering, we have conducted experiments using the "Hollywood Human Actions dataset" database [13], consisting of 32 movie clips. From the total of 32 movie clips we used only 20 of them in our experiments. The movie clips that have been excluded are a) 4 grayscale movies, as the NMI calculation is performed on colour facial images [6], and b) 8 more movie clips, where either the face detection algorithm returned only a few facial images, or the images that were returned belonged to one actor. Faces were detected in every movie clip video frame [6]. The number of facial images that the detection algorithm has provided us ranges between 52 and 243 images per movie clip, typically over 100 images per movie clip. In all experiments the images were scaled, in order to have the same size, considering all the detected facial images of the movie clip and using a mean bounding box as in [6]. The different actors that have been tracked in each movie clip range between 2 and 10 distinct individuals. False face detections have been manually excluded.

The metric used to evaluate the success of a clustering algorithm is the F-measure based on a combination of precision and recall measures [14]. Precision, in face clustering, is calculated as the ratio of correct instances to the total number of instances that actually belong to the cluster, while recall is calculated as the ratio of the correct instances to the total number of instances that the algorithm returned to belong in the cluster. Let S represent a set and $C = C_1, ..., C_n$ be a

TABLE I
FACE CLUSTERING RESULTS FOR [13]

Movie	λ	T	Clusters (Split)	Clusters (Merge)	F-measure
Bringing out the dead (3)	0.0095	0.7	6	3	87.31 %
Dead poets society (7)	0.0045	0.3	10	8	98.25 %
Gandhi (4)	0.0085	0.7	5	3	97.62 %
Kids (5)	0.0045	0.6	8	4	90.56 %
Lost highway (10)	0.003	0.7	15	8	92.70 %
Mission to Mars (4)	0.007	0.7	8	4	87.69 %
The pianist (5)	0.0096	0.3	7	6	97.20 %
Pulp fiction (2)	0.02	0.3	3	2	98.52 %
I am Sam (3)	0.01	0.7	4	2	93.69 %
Erin Brockovich (4)	0.006	0.53	6	4	100 %
Lord of the rings (4)	0.0075	0.7	6	3	96.00 %
Butterfly effect (6)	0.007	0.3	6	5	89.23 %
As good as it gets (3)	0.009	0.3	6	4	82.23 %
Big fish (9)	0.005	0.53	14	8	76.94 %
Forest Gump (6)	0.006	0.3	8	5	98.63 %
The Godfather (3)	0.0015	0.3	3	3	100 %
American beauty (6)	0.0075	0.3	5	5	93.52 %
Crying game (3)	0.0095	0.7	5	3	88.83 %
Graduate (2)	0.008	0.7	4	2	98.21 %
Indiana Jones and the last crusade (3)	0.0085	0.394	5	3	90.21 %

clustering of S with $C_i \cap C_j = \emptyset$ for $i \neq j$ and $i, j \in [1, ..., n]$. Moreover, let $C^* = C_1^*, ..., C_m^*$ be a representation of human reference clustering of S (i.e. the ground truth). Ground truth was constructed by human observation, as a compulsory way to evaluate the robustness of the algorithm. Then, precision of a cluster j with respect to cluster i can be defined as:

$$prec(i,j) = \frac{|C_j \cap C_i^*|}{|C_j|}, \tag{13}$$

while the recall of a cluster j with respect to cluster i is defined as:

$$rec(i,j) = \frac{|C_j \cap C_i^*|}{|C_i^*|}. \tag{14}$$

A high precision value means that most of the cluster images were correctly clustered, but we may have not found all the images that belong to that cluster (which would imply low recall value). High recall value means that we have not missed many images that belong to a cluster, but we may have a lot of facial images that do not belong to this cluster (which implies low precision value). Finally, the F-measure is defined as:

$$F(i,j) = 2 \times \frac{prec(i,j) \times rec(i,j)}{prec(i,j) + rec(i,j)}, \tag{15}$$

while the overall F-measure is given by:

$$F = \sum_{i=1}^{m} \frac{|C_i^*|}{|S|} \times \max_{j=1,\dots,n} F(i,j). \tag{16}$$

It is obvious that best results are obtained when there is a perfect match between human reference and the clustering output. Table I presents the results of the proposed clustering algorithm. The number in parenthesis, next to the name of each movie clip, represents the manually selected number of distinct actors in a movie clip (ground truth). We see that the F-measures obtained are very good, better than the ones proposed in [6]. The average F-measure was found to be $F = 92.87\%$. All images that have been detected vary in scale, pose and expression, as can be seen in Figure 2, providing evidence for the robustness of the algorithm.

By analysing a specific movie clip called “Forest Gump”, we have extracted a total of 184 detections, which belong to 6 different actors. When we applied the algorithm in order to split the facial images into clusters, we ended up with 8 facial image clusters, while the F-measure was 79.81%. Then, merging was applied and the facial image clusters were reduced to 5, giving an F-measure of 98.63%. Figure 2 illustrates one of the clusters of this movie clip.

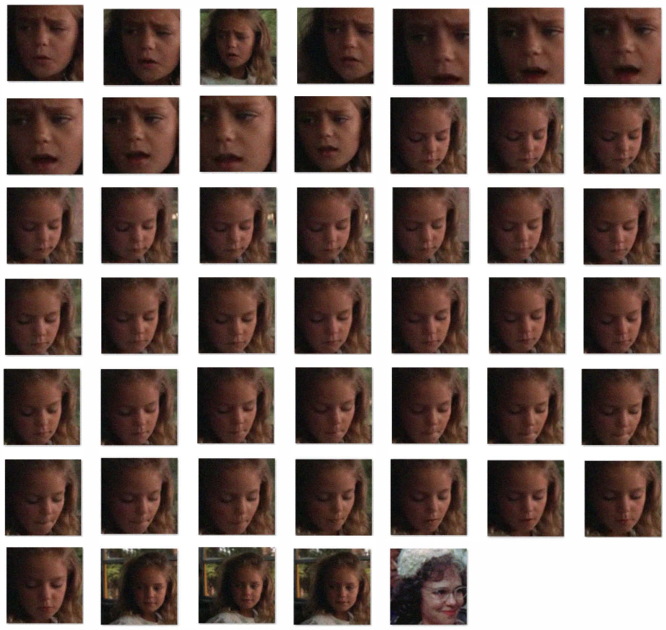

Fig. 2. A cluster from the movie clip “Forest Gump”.

IV. Conclusion

We have presented a novel algorithm for clustering facial images originated from movies. It is important to remark that face clustering in videos is a difficult task, due to the significant illumination, pose and facial expression variations. The technique of first partitioning a set of facial images and then deciding if some of the clusters should be merged appears to yield satisfactory results, in terms of the F-measure performance metric.

In the future, we aim to improve the proposed face clustering method, by testing different criteria for the split and merge steps. We shall also focus our efforts on merging clusters more efficiently, for example by attempting to merge more than two clusters at a time. Moreover, we shall try to find a way in order to handle false detection results within the framework (i.e. with a special cluster where all false detections can be placed in).

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme(FP7/2007-2013) under grant agreement no 248434 (MOBISERV)

References

[1] S. Eickeler, F. Wallhoff, U. Iurgel, G. Rigoll, "Content based Indexing of Images and Videos using Face Detection and Recognition Methods", IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP), Salt Lake City , Utah 2001.

[2] M. Everingham and A. Zisserman, "Automated person identification in video," Proc. of the 3rd International Conference on Image and Video Retrieval (CIVR2004), vol. 1, pp. 289-298, 2004.

[3] C. Czirjek, N. O'Connor, S. Marlow, N. Murphy, "Face Detection and Clustering for Video Indexing Applications", Advanced Concepts for Intelligent Vision Systems (Acivs), Ghent, Belgium, September 2003.

[4] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y. W. Teh, E. Learned-Miller, and D. A. Forsyth, "Names and faces in the news," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR'04). IEEE, 2004, vol. 2nd, pp. 848-854.

[5] A. W. Fitzgibbon and A. Zisserman, "On affine invariant clustering and automatic cast listing in movies", European Conference on Computer Vision (ECCV), vol. 3, pp. 304 - 320, Springer-Verlag, 2002.

[6] N. Vretos, V. Solachidis, and I. Pitas, "A Face Tracker Trajectories Clustering Using Mutual Information," in Multimedia Signal Processing, 2007, MMSP 2007, IEEE 9th Workshop on, 2007, Crete, 1-3 October.

[7] F. Bach and M. Jordan, "Learning Spectral Clustering," Proc. 17th Advances in Neural Information Processing Systems (NIPS '04), 2004.

[8] S. E. Schaeffer, "Graph clustering", Computer Science Review, 1 (2007), pp. 2764.

[9] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation" , Proc. IEEE Conf. Computer Vision and Pattern Recognition, June 1997.

[10] Zengyou He, Xiaofei Xu, and Shengchun Deng, "K-anmi: A mutual information based clustering algorithm for categorical data", 2005.

[11] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering", In Advances in Neural Information Processing Systems 17, 2005.

[12] M. Fiedler, "Algebraic connectivity of graphs, Czechoslovak Mathematical Journal, vol. 23, no. 98, 1973 , pp. 298305.

[13] Ivan Laptev and Marcin Marszałek and Cordelia Schmid and Benjamin Rozenfeld, "Learning Realistic Human Actions from Movies", IEEE Conference on Computer Vision & Pattern Recognition, 2008.

[14] Stein, B., S. M. Eissen, F. Wissbrock, "On Cluster Validity and the Information Need of Users", Proc. 3-rd IASTED Intern. Conf. on Artificial Intelligence and Applications (AIA'03), Acta Press, 2003, pp. 216221.