Image and Vision Computing 29 (2011) 693–705

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

A mutual information based face clustering algorithm for movie content analysis ☆

N. Vretos *, V. Solachidis [1], I. Pitas [1]

Department of Informatics, University of Thessaloniki, Thessaloniki 54124, Greece

ARTICLE INFO

Article history:
Received 13 December 2010
Received in revised form 15 July 2011
Accepted 29 July 2011

Keywords:
Face clustering
Mutual information
Normalized cuts
Spectral graph analysis
Image processing

ABSTRACT

This paper investigates facial image clustering, primarily for movie video content analysis with respect to actor appearance. Our aim is to use novel formulation of the mutual information as a facial image similarity criterion and, by using spectral graph analysis, to cluster a similarity matrix containing the mutual information of facial images. To this end, we use the HSV color space of a facial image (more precisely, only the hue and saturation channels) in order to calculate the mutual information similarity matrix of a set of facial images. We make full use of the similarity matrix symmetries, so as to lower the computational complexity of the new mutual information calculation. We assign each row of this matrix as feature vector describing a facial image for producing a global similarity criterion for face clustering. In order to test our proposed method, we conducted two sets of experiments that have produced clustering accuracy of more than 80%. We also compared our algorithm with other clustering approaches, such as the k-means and fuzzy c-means (FCM) algorithms. Finally, in order to provide a baseline comparison for our approach, we compared the proposed global similarity measure with another one recently reported in the literature.

1. Introduction

Face clustering is a very important task for movie semantic extraction. It can contribute in many ways, like determining the principal actors or the creation of database references or dialog detection and many others. Moreover, face clustering can be used for unsupervised training of face recognition algorithms and in general as a preprocessing step in any human based image and video processing tasks, so as to create a human wise categorization of the data.

Facial image clustering, put together facial images that belong to the same person by employing a certain image similarity criterion. Let P be a set of facial images. A clustering $\mathcal{C}=\{C_i|C_i\subseteq P\}$ is a division of P into facial image clusters C_i, for which the following conditions hold: $\cup_{C_i\in\mathcal{C}} C_i=P$ and $\forall C_i,C_j\in\mathcal{C}: C_i\cap C_{j\neq i}=\varnothing$. Ideally, the clustered facial images should belong to the same person. Face clustering is a very important application and can contribute in many ways to semantic movie analysis, e.g., for determining the movie cast or for assisting automatic dialog detection. Until now, few face clustering algorithms have been reported in the literature [1–4].

Face recognition and face clustering are two different tasks: in face recognition, we assume that we have a known number of persons and a training facial image database, consisting of certain labeled facial images per person. This database is used for training a face recognition classifier. Then, if we have a test video, each facial image extracted from a video frame can be tested by the already trained face recognition classifier and the best matching person id (or rather a list of best matching people ids) is returned. In face clustering, the number of persons appearing in a video clip or movie is unknown and there is no training facial image database. Therefore, no training is possible. The face clustering goal is entirely different from that of face recognition: given a number of video frames containing facial images, we have to find the unknown number of persons appearing therein, based on facial image similarities. Both face recognition and face clustering may share certain tools (e.g. image similarity measures, face representation methods), but are different in many aspects in terms of goals, methodology (training/no training) and performance metrics. Although, a great amount of work has been conducted on face recognition, face clustering is a rather novel topic with few publications in the literature so far [1–4]. In [2] the authors have proposed an approach for face clustering in video that involves the so called Joint Manifold Distance (JMD). Therein, the authors propose a method, where each subspace represents a set of facial images of the same person detected in consecutive frames. The clustering algorithm, uses a facial video sequence to sequence distance and follows an agglomerative strategy. Another distance metric for clustering and classification algorithms, called Affine Invariant Distance Measure (AIDM) was proposed in [3]. This distance function, which is invariant to affine transformations, is used in combination with partitioning-based algorithms for face clustering. In [4], Foucher et al. recommended a face clustering method based on face detection and tracking

☆ This paper has been recommended for acceptance by Ioannis A. Kakadiaris.

* Corresponding author. Tel./fax: +30 2310996304.
E-mail addresses: vretos@aiia.csd.auth.gr (N. Vretos), pitas@aiia.csd.auth.gr (I. Pitas).
[1] Tel./fax: +30 2310996304.

0262-8856/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2011.07.006

and use several spectral graph techniques for classification. Finally, in our previous work [1], we have proposed a mutual information (MI) based technique for face clustering by constructing a similarity matrix based on the image intensities and have clustered this similarity matrix by means of a fuzzy c-means classifier. The motivation to employ MI comes from its numerous uses as an image similarity measure in other image analysis tasks, e.g. in medical image registration [5], shot cut detection [6], and object tracking [7].

In many applications involving facial images, color spaces are exploited in order to better characterize the facial features. In [8], a specific color space is used for face recognition. In the proposed method, the HSV color space of a facial image (more precisely, only the hue and saturation channels) is used, in order to calculate the mutual information similarity matrix of a set of facial images. We make full use of the similarity matrix symmetries, so as to lower the computational complexity of the mutual information calculation. Thereafter, we assign each row of this matrix as feature vector describing a facial image for producing a global similarity criterion for face clustering. Finally, spectral graph clustering of the global similarity matrix is used to perform clustering.

Spectral graph clustering has been used in image segmentation [9], object recognition [10] and graph-matching [11]. In [12], Carcassoni and Hancock use a coarse-to-fine detail approach, in order to provide a more robust graph clustering process and to overcome problems that arise from spurious graph nodes and edges. In our case, the facial images in a facial image set P can be considered as nodes in a similarity graph, whose edge weights are the facial image similarity. Thus, spectral graph can provide node (i.e. facial image) clustering. As will be demonstrated later on, spectral graph analysis outperforms other clustering methods in face clustering.

The novelty of our approach is primarily in the use of hue and saturation in the calculation of the MI in assessing facial color image similarity, versus the more commonly used image intensity MI [1]. Thus, the proposed method is proven to be robust when, we have facial pose and illumination variations. Moreover, we use a novel feature vector that describes the global similarity of a facial image to the rest of the facial images. This fact provides extra robustness to the proposed method. Finally, spectral graph clustering is applied on the global similarity matrix, which provides superior performance than competing techniques, e.g. k-means or FCM used in [1]. It also outperforms other methods that are used in image registration, mainly due to the fact that such methods are much simpler with respect to light variations and pose and, thus, inappropriate for the face clustering task.

The remainder of this paper is organized as follows: facial color image mutual information and its normalized version (to be used as facial image similarity measures) are presented in Section 2. In Section 3, we present face clustering using N-cuts. In Section 4, we show the face clustering performance metrics and experimental results on two test cases: a) the XM2VTS facial video database [13] and b) another video database coming from extracts of six commercial movies. In the same section, we provide a "baseline" comparison of the employed similarity criterion (i.e. the hue/saturation MI) to another newly developed image similarity criterion [14]. Finally, conclusions are drawn in Section 5.

2. Mutual information for color facial image clustering

Many image similarity measures have been proposed in recent literature [15–21]. An extensive survey of f-measurements and various entropy measures, e.g. the Rèny and Tsalis entropy, are presented in [15]. Other image similarity measures, like the Kullback–Leibler divergence [16] can be used as well. Recent approaches [17–21] to image registration use the mutual information (MI) measure that is proven to be robust under cropping and small illumination perturbations.

The mutual information of two random variables is defined as:

$$I(X,Y) \triangleq \sum_{x\in X}\sum_{y\in Y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}, \tag{1}$$

where $p(x,y)$ is their joint pdf and $p(x)$, $p(y)$ are their marginal pdfs. Typically, X, Y represent the image intensity of two different images. The entropy of a random variable X is defined as:

$$H(X) = -\sum_{x\in X} p(x)\ \log p(x). \tag{2}$$

Likewise the joint entropy of two random variables X and Y is defined as:

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

There are several ways of normalizing the mutual information between different pairs of images [22]. The normalized version of mutual information used in this paper is defined as in [17]:

$$N_{MI}(X,Y) \triangleq \frac{H(X)+H(Y)}{2H(X,Y)}, \tag{4}$$

N_{MI} takes values in the domain [0,1]. In [22], Studholme et al. have shown that this version of the normalized mutual information is less sensitive to the size of the overlapping image regions in image registration. A detailed presentation of the aforementioned entropy and mutual information calculation can be found in [23].

In the case of color facial images, we shall use the HSV color space for checking similarity and, in particular, the hue H and saturation S components, which are proven to be robust under illumination changes, in comparison to image intensities [24,25]. In [26], Sobottka and Pitas have shown that face colors occupy a certain region of the HSV color domain. Furthermore, it is proven that, at a specific region of the HS domain, H is the most informative channel [27] for facial colors. Therefore, we employ only the H, S channels of two facial images having hue and saturation values H_1, S_1, H_2, S_2 respectively.

It can be easily shown that the 4D normalized MI is given by:

$$N_{MI}(H_1,S_1,H_2,S_2) = \frac{H(H_1)+H(S_1)+H(H_2)+H(S_2)}{2\cdot H(H_1,S_1,H_2,S_2)}. \tag{5}$$

Let us suppose that the histograms $\hat{p}(h_1)$ and $\hat{p}(h_2)$ to be used in Eqs. (2), (5) have N bins, while $\hat{p}(s_1)$ and $\hat{p}(s_2)$ have M bins. The 4D histogram estimating $\hat{p}(h_1,s_1,h_2,s_2)$ to be used in Eq. (5) has dimensions $N\times M\times N\times M$ and can be found as follows. Let X_1, X_2 be two facial color image regions of interest (ROIs) of size $H\times W$ pixels produced by a face detector/tracker. We transform them in the HSV color space and calculate the 4D joint histogram:

$$\hat{p}\,(h_1,s_1,h_2,s_2) = \frac{1}{H\cdot W}\cdot|\{(k,l)\in[1,H]\times[1,W]\,/\,H_1(k|l) = h_1 \tag{6}$$
$$and\ S_1(k,l) = s_1\ and\ H_2(k,l) = h_2\ and\ S_2(k,l) = s_2\}$$

where $|\cdot|$ denotes set cardinality and $H_1(k,l)$, $S_1(k,l)$, $H_2(k,l)$, $S_2(k,l)$ are the hue and saturation values for image X_1 and X_2 at pixel (k,l), respectively. Then, $\hat{p}(h_1)$, $\hat{p}(s_1)$, $\hat{p}(h_2)$, $tp(s_2)$, $\hat{p}(h_1,s_1,h_2,s_2)$ and Eqs. (2), (3) are used in calculating Eq. (5). The facial images X_1, X_2 in Eq. (6) must have the same size of $H\times W$ pixels, which is not always true, since face detectors typically produce facial regions of varying size. In order to overcome this problem, we calculate a mean bounding box from the face detector/tracker results on a particular video and scale all facial images to this size. After several experiments, we have concluded that this is the best way to solve the scaling/cropping problems. Other approaches, e.g., scaling each pair of facial image ROIs towards the biggest or the smallest bounding box of the

face detector/tracker results, produced inferior clustering performance. This superior performance can also be attributed to the way we use the ensuing similarity matrix. As will be explained in more detail in a subsequent section, the similarity matrix row i represents the similarity of a facial image i with all other images. By proposing an overall mean bounding box, we have a more informative similarity matrix, due to the fact that facial image resizing problems are minimized. If anisotropic scaling of the facial image ROI is needed, then we need to take care of the aspect ratio, in order to avoid disturbing the facial images by stretching. To do so, we first calculate the aspect ratio $\overline{W}/\overline{H}$ of the mean bounding box, where $\overline{W}$ and $\overline{H}$ are its width and height, respectively. We need to preserve this ratio in each facial image ROI while scaling towards the mean bounding box. To do so, we preserve the aspect ratio $\frac{W}{H} = \frac{\overline{W}}{\overline{H}}$ of the width W and height H of each facial image ROI, by padding along the x or y axes with adequate video frame rows and columns bordering the bounding box. The dimension that is padded is always the smaller one in length. Let $R=\{(x_1,y_1),(x_2,y_2)\}$ be a bounding box defined by its upper-left corner (x_1,y_1) and the down-right one (x_2,y_2). Then $W=x_2-x_1$, $H=y_2-y_1$ denote its width and height respectively. If $\frac{W}{H} > \frac{\overline{W}}{\overline{H}}$, the resulting bounding box must have a new height $H' = \frac{W\cdot\overline{H}}{\overline{W}}$, resulting to a new bounding box $R' = \left\{\left(x_1, y_c - \frac{H'}{2}\right), \left(x_2, y_c + \frac{H'}{2}\right)\right\}$, where y_c denotes the vertical coordinate of the initial bounding box center. A similar analysis can be done in the case where $\frac{W}{H} < \frac{\overline{W}}{\overline{H}}$, resulting to a new bounding box $R' = \left\{\left(x_c - \frac{W'}{2}, y_1\right), \left(x_c + \frac{W'}{2}, y_2\right)\right\}$, with x_c being the horizontal coordinate of the initial bounding box center.

The facial image ROI detection and tracking are used in a way that no manual initialization is needed. Since the paper does not focus on face detection and tracking, which have been heavily researched, we follow the process described in [28]. It consists of running the face detector every n video frames (typically $n=5$) and then, if the face detection is successful, the face tracker tracks the detected faces for the next $n-1$ frames. Then the face detector is launched again. A series of criteria are used for the re-initialization of the tracker [28].

Finally, we have noticed that, in several cases, either the face detector or the tracker produce facial images, which are either cropped or containing too much background, as shown in Fig. 1. In order to solve such problems, for every facial image pair produced by the face detector/tracker, we calculate the mutual information of two facial ROIs for different bounding box sizes and we take the maximum MI value achieved. To this end, we start varying the bounding box size from 80% to 120% of the initial mean bounding box with a step of 5% along both axes, using uniform facial image ROI cropping or expansion, respectively.

Fig. 1. Face images where faces have the same actual size but not the same bounding box.

Using the aforementioned definition (Eq. (5)) for N facial images, X_i, $i=1,..,N$, we construct the $N\times N$ similarity matrix $\mathbf{M} = \{N_{MI}(X_i,X_j), i,j\in[1,N]\}$. Such a similarity matrix is visualized in Fig. 2. The brighter boxes correspond to higher similarity of the facial images of various appearances of the three actors involved in this video. It is a symmetric matrix along the main diagonal due to the fact that $N_{MI}(X,Y)=N_{MI}(Y,X)$, because $p(X,Y)=p(Y,X)$ for any X, Y random variables, as can be easily proved by applying the Bayes rule in the definition of the joint probability [29].

2.1. The use of face tracking heuristics in the facial image similarity matrix

In order to integrate some a priori face detection and tracking information in face clustering, we assume that:

1. the used face tracker tracks faces correctly;
2. the facial image of each person appears only once in each video frame.

When we track a face, we assume that the tracked facial image ROIs define a so called *face appearance*. The tracking results provide a link between facial image ROIs of consecutive frames. We make use of this information to assign a facial image to a specific face appearance. That is, as long as the tracker follows the same ROI (i.e. no tracker re-initialization is needed), we assert that these facial images belong to the same face appearance. In this case, we use the following similarity definitions:

$$M'_{ij} = \begin{cases} 1, & \text{if } (X_i, X_j) \text{ belong to the same face appearance} \\ 0, & \text{if } (X_i, X_j) \text{ belong to the same frame} \\ M_{ij}, & \text{if none of the above is true,} \end{cases} \tag{7}$$

where X_i and X_j denote facial image ROIs and M_{ij} the (i,j)-th element of matrix $\mathbf{M}$. This a priori information changes the structure of the similarity matrix, as shown in Fig. 3. The white squares along the diagonal of M' are due to facial image ROIs belonging to the same face appearance. The two black checkered boxes along the main diagonal are due to the fact that two facial image ROIs appear in the same video frames. Therefore, according to Eq. (7), their modified MI is zero. Naturally, these previously mentioned assumptions may fail, like in

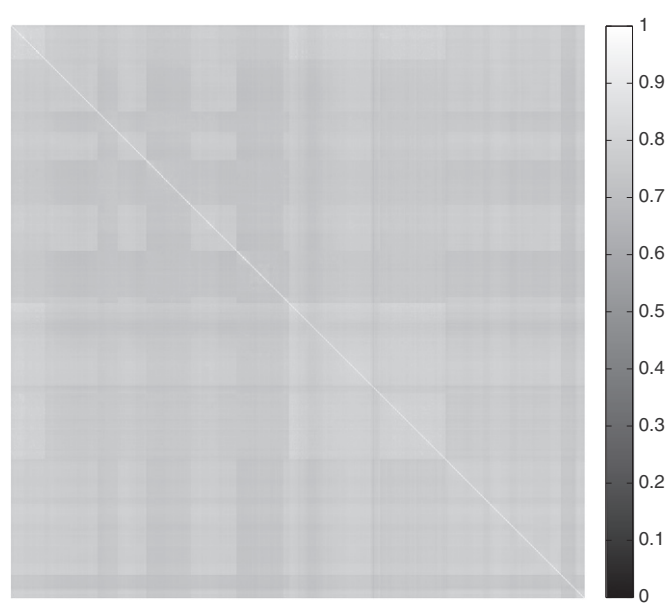

Fig. 2. Similarity matrix **M** for three different actors in 941 detection.

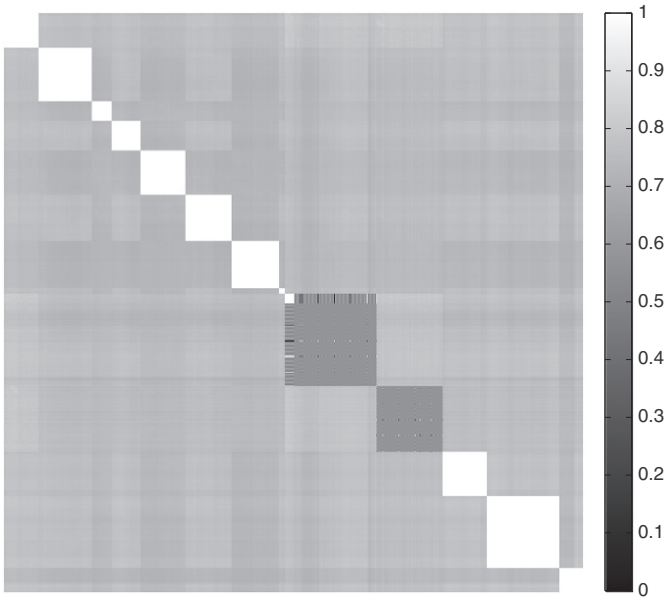

Fig. 3. Robust version M' of the same similarity matrix **M** as depicted in Fig. 2.

the case where multiple versions of a face appear in the same video frame (e.g. when a mirror is used or the same actor plays more than one role). However, such cases occur very seldom. When the tracker erroneously stops tracking one person and starts tracking him/her again after a while (e.g. after an occlusion), M' does not fully exploit the tracking information. However, even in this case, this is not a major problem, since the value M_{ij} is already high, if facial images X_i, X_j belong to the same person (but not to the same face appearance). Finally, we define the so called global similarity matrix **W**, whose elements W_{ij} contain global similarity information between facial images X_i, X_j:

$$W_{ij} \triangleq exp\left\{-\frac{d(i,j)}{\sigma^2}\right\}, \tag{8}$$

$$d(i,j) \triangleq \sqrt{\sum_{k=1}^{N} \left(M'_{ik} - M'_{jk}\right)^2}, \tag{9}$$

where σ is an appropriate chosen parameter. In our experiments we used several values of σ, with the value $\sigma = 0.1$ yielding the best clustering results.

Each row M'_i, $i = 1,..,N$ of matrix M' describes the similarity of the facial image X_i towards all others. If images X_i, X_j belong to the same person, it is expected that they will have the same similarity values $M'_{ik} \simeq M'_{jk}$, to all other images $k = 1,..,N$ with $k \neq i$ and $k \neq j$. In this case, their global similarity is $W_{ij} = 1$. If the images X_i, X_j belong to different persons, then in general $M'_{ik} \neq M'_{jk}$ for $k = 1,..,N$ with $k \neq i$ and $k \neq j$. Then $d(i,j)$ is large and thus W_{ij} tends to 0. Therefore, matrix **W** contains a more global image similarity information rather than that of matrix M' that contains similarities between pairs of facial images. Hence, **W** is expected to provide better clustering results, since two facial images belonging to the same person may be dissimilar (in the sense of MI image similarity), but they may both be similar to the same third image. As an illustrative example of this concept, we can consider a left 3/4 and right 3/4 pose of the same face, that may be quite distant in terms of MI similarity. These two facial images, though, have a big mutual information value with a frontal image which belongs to the same person. Thus, these two visually distant facial images are linked in the global similarity matrix **W**. In Fig. 4, we have plotted three matrix rows of the M' matrix, two from facial images belonging to the same person (Face1 and Face3) and one from facial images belonging to a different person (Face2). Ideally, the similarity values M'_{Face1k} and M'_{Face3k} should be identical and different from M'_{Face2k} for $k = 1,..,N$. However, this is not the case always. For instance, for $k = 371,..,447$, the values M'_{Face2k} are closer to M'_{Face3k} than M'_{Face1k} to M'_{Face3k}. However, Face2 and Face3 are still globally more distant. This is manifested by the fact that $W_{Face1Face3}$ is much greater than both $W_{Face1Face2}$ and $W_{Face2Face3}$. Thus, the face clustering algorithm using **W** will correctly cluster together Face1 and Face3 facial images.

3. Spectral graph clustering

The global similarity matrix **W** corresponds to a similarity graph, whose nodes $k = 1,..,N$ are the facial images to be clustered and the similarities W_{ij} corresponds to the graph edge weights. We use spectral graph theory, in order to cluster the similarity graph. To do so, we create the associated combinatorial Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$ of the similarity graph, where **D** is the diagonal matrix with $D_{ii} = \sum_j W_{ij}$. The eigenanalysis of matrix **L** provides useful information about the graph structure. It can be shown that **L** is a positive semi definite matrix, and therefore, all eigenvalues are not negative, a fact that will be used for faster eigenvalue calculation [29].

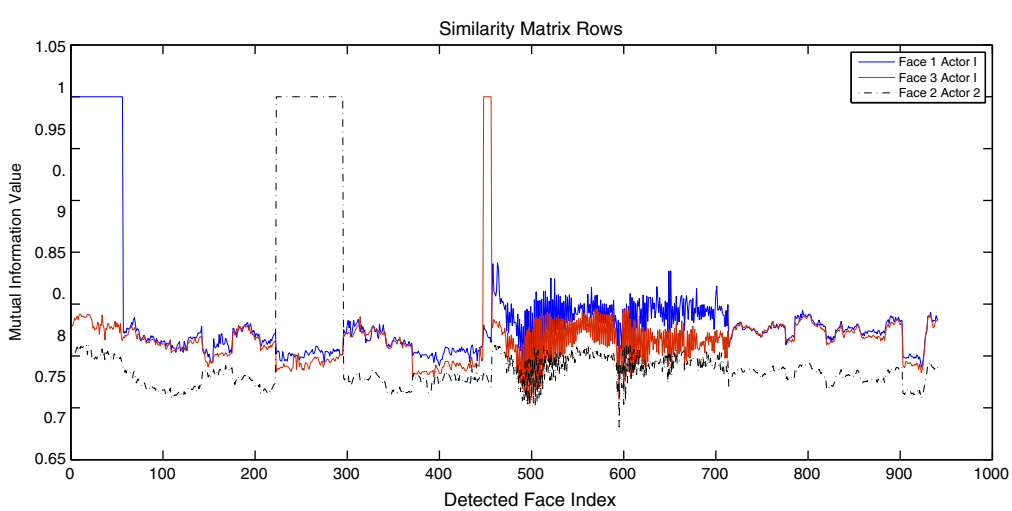

Fig. 4. Visualization of three rows of similarity matrix M'.

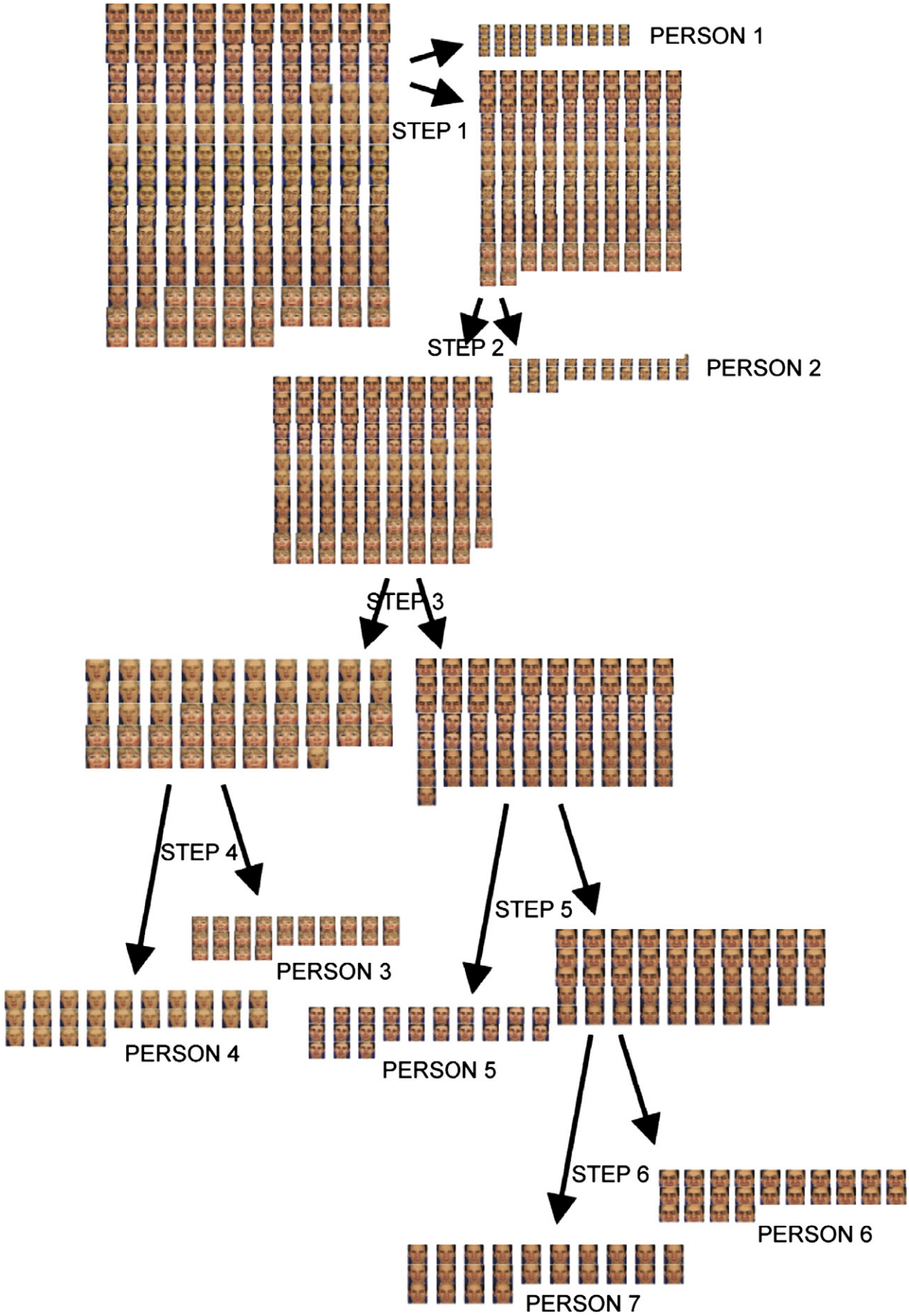

Fig. 5. The partitioning process of the facial image similarity matrix of the XM2VTS image data set.

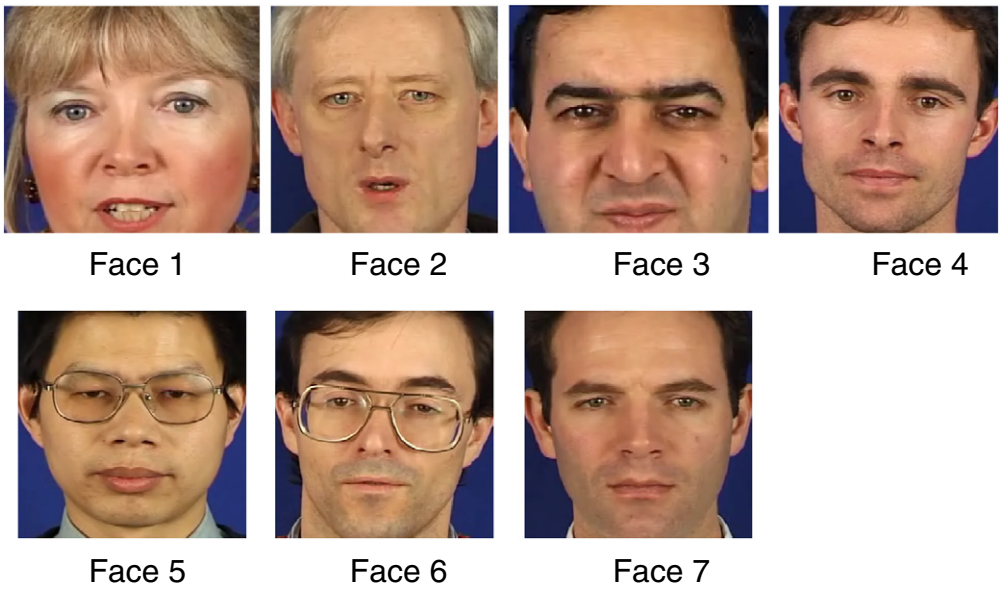

Fig. 6. XM2VTS facial images.

3.1. Normalized cut clustering

In order to cluster the facial images, we use the normalized cuts method proposed in [9]. More precisely, we use the recursive two-way N-cut algorithm [9], which, in summary, consists of the following steps:

1. Calculate the matrices $\mathbf{W}$, $\mathbf{D}$ and $\mathbf{L}$ as previously described.
2. Calculate the median $Med(\mathbf{W}) \triangleq med\{med\{W_{ij}, i = 1, \ldots, N\}, j = 1, \ldots, N\}$. If $Med(\mathbf{W})bT$, then continue recursion else Goto step 6.
3. Solve $\mathbf{Lv} = \lambda\mathbf{Dv}$.
4. Use the eigenvector with the second smallest eigenvalue to bipartition the global similarity matrix $\mathbf{W}$ into $\mathbf{W}_1$ and $\mathbf{W}_2$ based on the signs of the eigenvector coefficients.
5. Recursively repartition the subgraphs from step 1, with $\mathbf{W} = \mathbf{W}_1$ and $\mathbf{W} = \mathbf{W}_2$.
6. Return the facial image clusters extracted from all recursions.

The aforementioned algorithm will result in a subdivision of the initial graph into disjoint clusters, according to a predefined threshold T. If the median of these entries is big, then the facial images participating to the creation of the global similarity matrix belong to the same person. In [9], many other approaches are proposed in order to stop recursion. It has been noted that the threshold value is highly sensitive to facial pose and illumination changes. In such cases, the dispersion of the facial image class corresponding to one person is very large to be accommodated by the methods employed in this paper (i.e. global similarity matrix, use of *HS* color space) and is prone to produce more clusters for the same person. Although, it was not possible to determine this threshold in an unsupervised way, this fact was not a major drawback to the whole framework. We can also provide an interval for the threshold T, where the clustering process performs better for all experiments we have conducted. Furthermore, the N-cut clustering algorithm has much less computational complexity than then calculation of the global similarity matrix $\mathbf{W}$. Once we have estimated $\mathbf{W}$ using Eqs. (5), (8) and (9), we can easily cluster it for different values of the threshold T. Each clustering run costs merely few seconds (typically 1–10 s) on a Pentium 4 3.0/GHz with 1 GB of memory. In Fig. 5, we can see the different steps of facial clustering for a particular value of threshold T. We see that, initially, all facial images corresponding to 7 distinct persons (ideally each forming one cluster) contribute in the similarity matrix *thbfW*. In the first iteration step, the facial images of a single person already create their own cluster. In the second step, another cluster is created corresponding to a single person. At step 3, two clusters are formed, one containing facial images of two persons and another one containing facial images of three persons. These clusters are split in one person clusters in subsequent steps.

As we have to calculate and solve the eigenvalue problem $\mathbf{Lv} = \lambda\mathbf{Dv}$ at every recursion, the algorithm is expected to be relatively slow. In order to tackle this problem, we use the Lanczos method, which is derived from the well known Implicitly Restarted Arnoldi (IRA) method [30]. This is possible due to the fact that the eigenproblem solution errors do not influence the sign of the eigenvector coefficients that are subsequently used to bi-partition $\mathbf{W}$.

Finally, as mentioned before the time complexity of the similarity matrix calculation is the task consuming most of the time in the whole framework. In order to give some insight of the time consumed in a real movie, we have calculated that we need approximately 0.01 s in order to calculate the similarity between two images. For a real movie

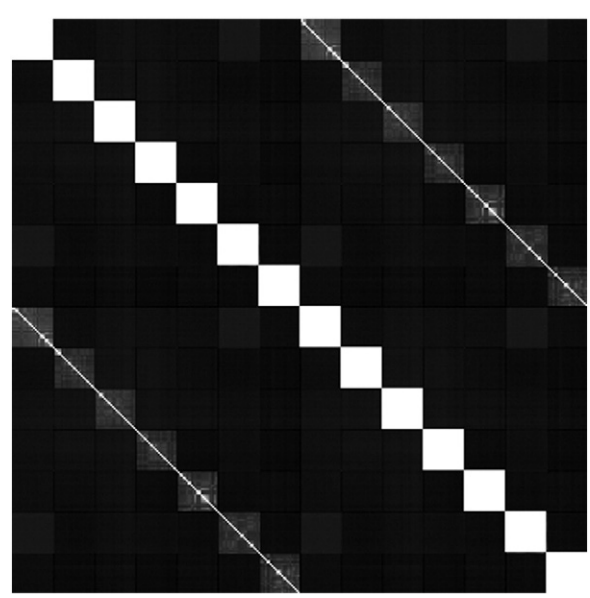

Fig. 7. Facial similarity martix for the XM2VTS database.

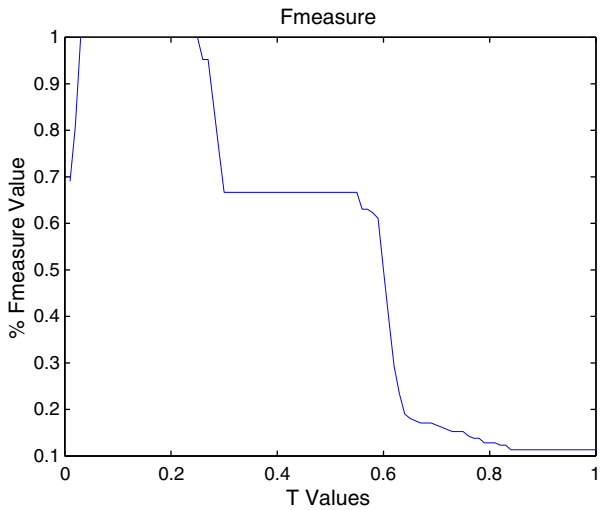

Fig. 8. *F*-measure for different clustering thresholds *T* for the XM2VTS database.

we have approximately 10^4 detections. Overall we need to calculate $\frac{10^8}{2}$ similarities. This results in approximately 6 days to calculatethe similarity matrix.

4. Experimental results

In this section, we shall describe first the evaluation framework of our algorithm. A very good review for several clustering performance metrics is presented in [31]. In our case, we have chosen the *F*-measure, which combines the precision and recall measures used in information retrieval. Let *P* represent a facial images set and let $C=\{C_1,\ldots,C_K\}$ be a clustering of *P*. Moreover, let $C^*=C_1^*,\ldots,C_L^*$ be the cluster ground truth. Then the recall of cluster *j* with respect to cluster *i*, $r(i,j)$ is defined as $\frac{|C_j\cap C_i^*|}{|C_i^*|}$, where $|\cdot|$ is the set cardinality. The precision of cluster *j* with respect to cluster *i* $p(i,j)$, is defined as $\frac{|C_j\cap C_i^*|}{|C_j|}$. The *F*-measure combines both values as follows:

$$F_{i,j}=\frac{2}{\frac{1}{p(i,j)}+\frac{1}{r(i,j)}}=\frac{2p(i,j)r(i,j)}{p(i,j)+r(i,j)}. \tag{10}$$

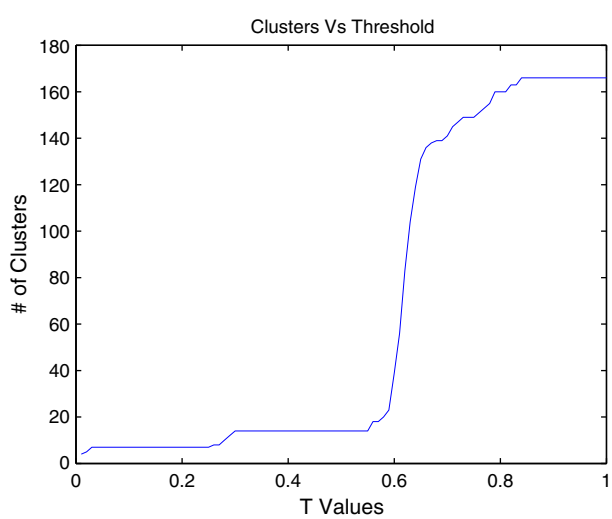

Fig. 9. Number of clusters produced for different clustering thresholds *T* for the XM2VTS database.

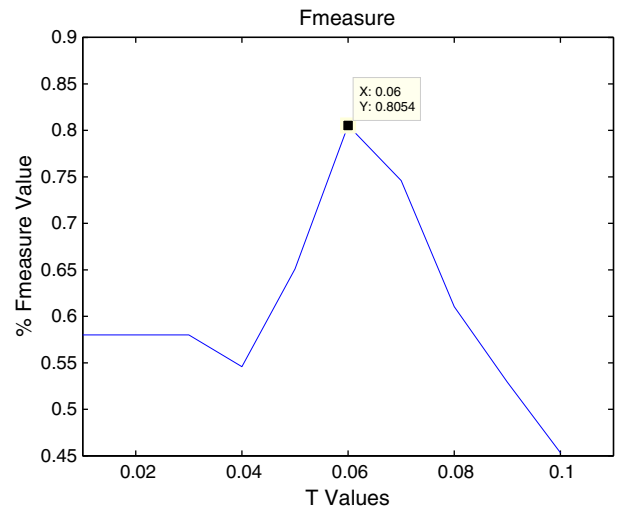

Fig. 10. *F*-measure for different clustering thresholds *T* for facial images from the movie "Two weeks notice".

The overall *F*-measure of clustering then is given by:

$$F=\sum_{i=1}^{L}\frac{|C_i^*|}{|P|}\cdot\max_{j=1,\ldots,K}\left\{F_{i,j}\right\}. \tag{11}$$

We can easily note that a perfect fit between the face clustering results and the ground truth leads to an *F*-measure score $F=1$. The *F*-measure uses ground truth information. Other measures, like the Dunn and Davies–Bouldin indices [31], can be used, in order to evaluate the cluster compactness, when ground truth is unavailable. We believe that cluster compactness provides limited information in the case of face clustering, since facial images of the same person can be quit distant visually, as well as in terms of mutual information (e.g. due to pose change), but close to each other semantically. This results in a low cluster compactness. For this reason, measures based on the clusters compactness are inappropriate for our experiments. We have seen that, for practically the same *F*-measure, the cluster compactness varies considerably in different facial images sets.

We have conducted two sets of experiments for assessing the performance of the proposed algorithm. We have first tested it with

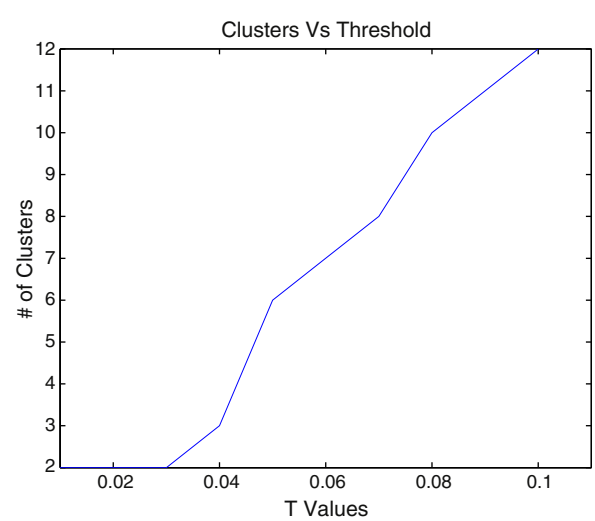

Fig. 11. Number of produced clusters for different clustering thresholds *T* for facial images of the movie "Two weeks notice".

Fig. 12. Samples of facial images with pose variations that are correctly clustered in one cluster.

the facial videos extracted from the XM2VTS database [13]. We have chosen two talking heads videos of 7 persons shown in Fig. 6, each video having 25 frames for each of the 7 persons (a total of 350 facial images). The faces were detected and subsequently tracked in these videos using the face detection/tracking algorithms described in [32]. The face detector/tracker produced ROIs of a mean size of 284×284 pixels. Fig. 7 visualizes the produced similarity matrix M'. Note that some facial images belonging to the same person but to different video clips are easily distinguished from others in the similarity matrix M', as can be seen from the off diagonal gray squares. The face clustering algorithm produced 7 separated clusters using a threshold T in the range $[0.04, 0.23]$. The resulting F-measure was 1. Thus, the face clustering was perfect. The fact that we employed relatively large facial images explains the required computation effort (approximately 4 h

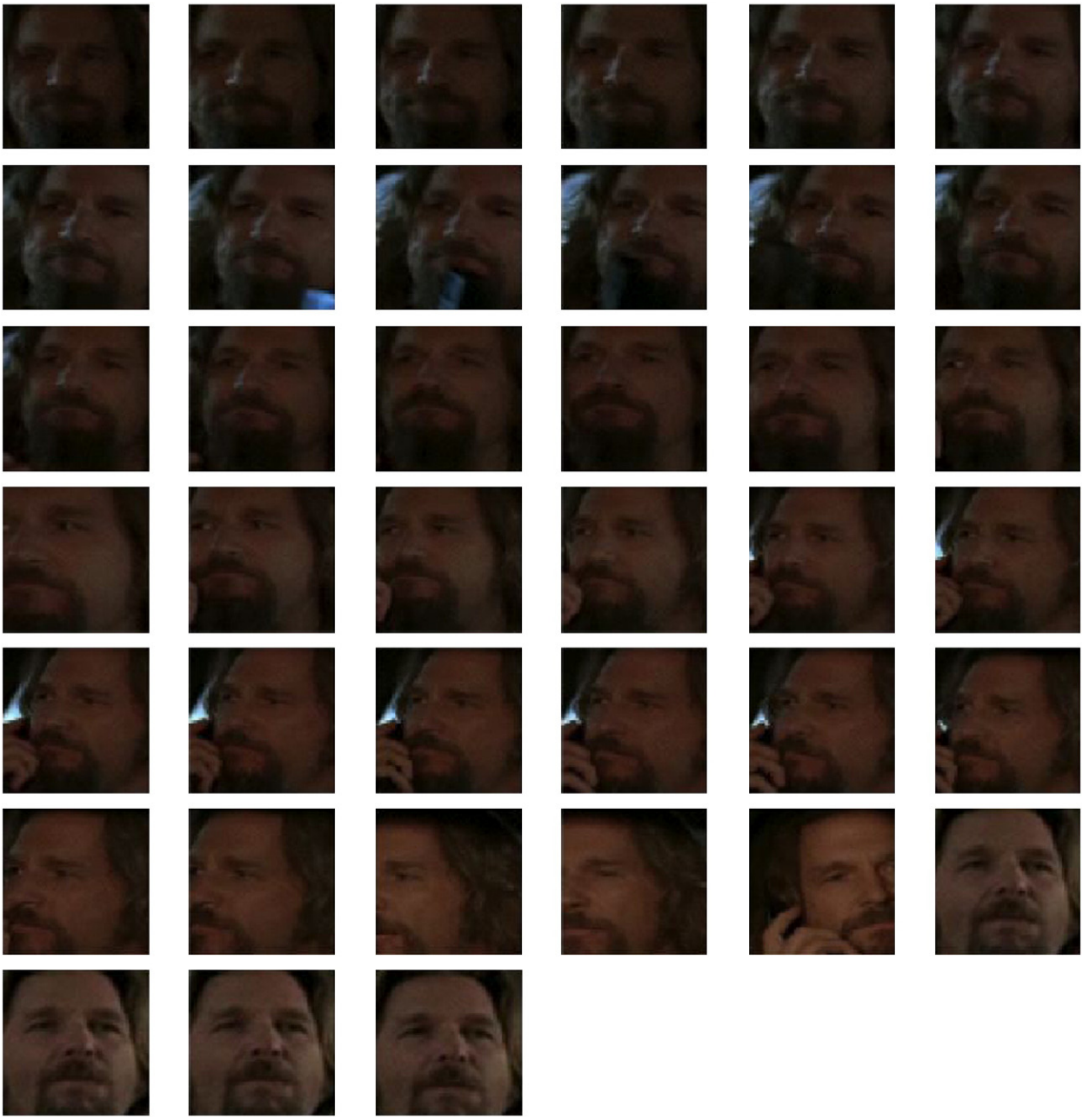

Fig. 13. Samples of facial images with illumination variations that are correctly clustered in one cluster.

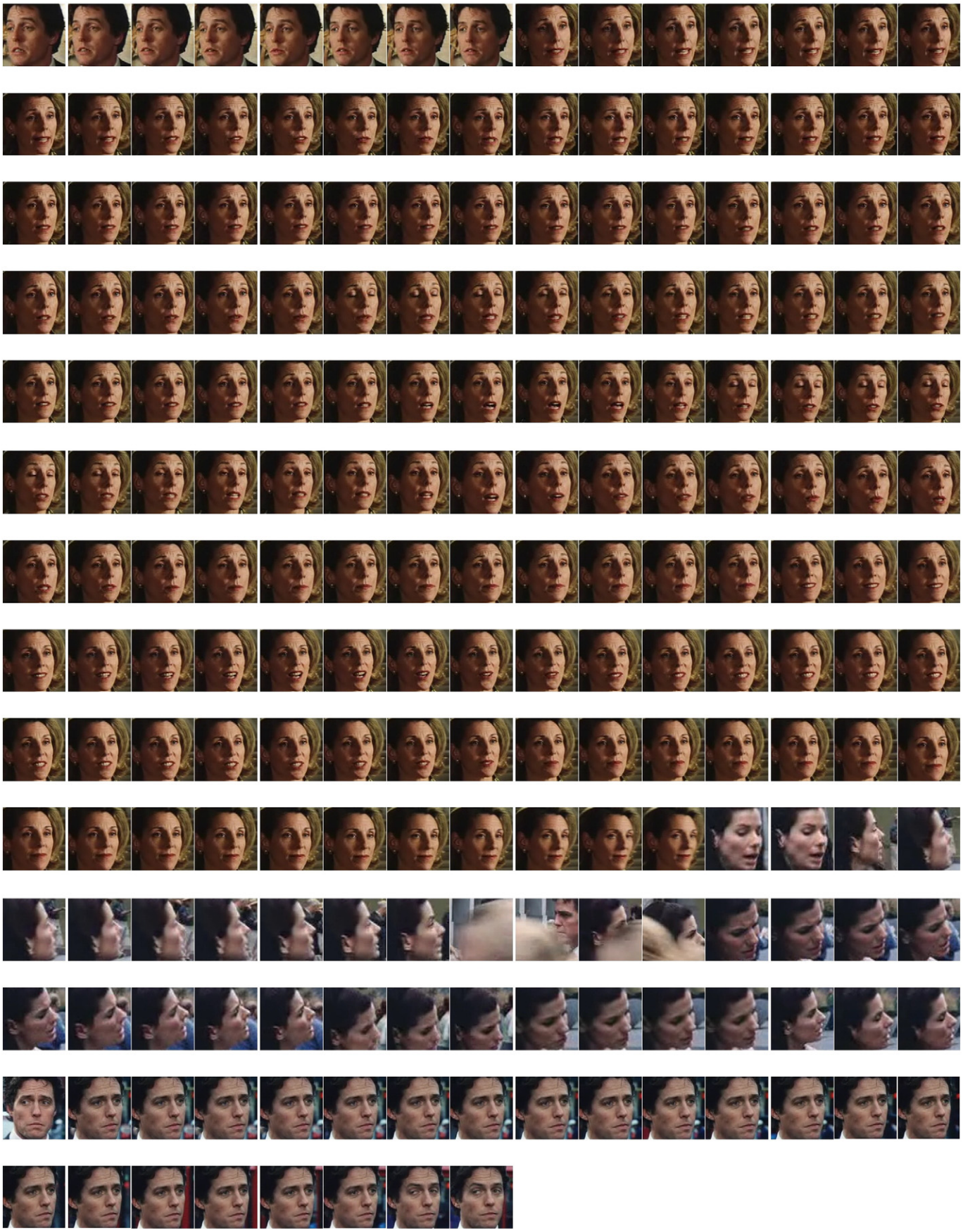

Fig. 14. One cluster resulting from the N-cut algorithm when using the similarity matrix M'.

on a Pentium 4 3.0/GHz with 1 GB of memory). Most of the computation time was consumed in the hue/saturation MI calculation. In Fig. 8, we can see the results of the F-measure for different clustering thresholds. For a certain range of thresholds $T \in [0.04, 0.22]$, we achieved perfect clustering ($F=1$) and the correct number of clusters $k=7$, as shown in Fig. 9.

The second experiment was conducted on 16 video clips that were parts of 16 movies. The last 10 movies were part of the Hollywood database [33]. We have manually created ground truth data for each movie clip. Approximately 2000 frames from each movie were used, totaling 34,536 video frames. Shot cut detection was performed first [6], in order to assist the face tracking algorithm as described in [28].

We worked on each video independently. We shall describe in detail only the results for one video, as the face clustering procedure was identical for all 16 videos. For the video "Two weeks notice", the face detection/tracking process produced a total of 15 face appearances (consecutive video frames containing faces) having in total 941 detected facial images. Three different actors appear in these facial images. We produced ground truth data consisting of four clusters, one for each actor, plus one extra "garbage bin" cluster containing the false detections of the detector/tracker. In this test case, the facial images had a mean bounding box of 114×114 pixels. The experiment required approximately 6.5 h to run on the previously mentioned computer. Figs. 10 and 11 show the F-measure and the number of produced clusters respectively, for different clustering thresholds, respectively. The best results were achieved for threshold $T=0.06$ ($F=80.54\%$, 7 clusters produced for 3 actors). We see that the face clustering results are sensitive to the threshold choice. This is expected, since we have large facial image similarity dispersion within each facial image class, e.g. due to pose variations and illumination changes, as can be seen in Figs. 12 and 13. Most of these variations are handled by using the global similarity matrix $\mathbf{W}$ instead of M', as discussed earlier. However, some pose changes are too large to be accommodated by using $\mathbf{W}$. In this case, it is easy to run the face clusteringfor several T and choose larger threshold values that lead to more clusters that are compact and correspond to particular poses of the same actor.

Fig. 15. One cluster resulting from the N-cut algorithm when using the similarity matrix $\mathbf{W}$.

Fig. 16. One cluster resulting from the *N*-cut algorithm when using the similarity matrix **M**.

We have also tested the assumption made in Section 2 that the use of **W** outperforms that of the M' matrix, due to the fact that it introduces a more global similarity criterion. Examples of facial image clusters produced by M' and **W** can be seen in Figs. 14 and 15, respectively. It is clearly shown in Fig. 14 that the produced cluster has many errors (i.e. facial images from different actors), while a robust cluster of one actor is displayed in Fig. 15. In both cases the *N*-cut algorithm was employed with the same threshold *T*. As a result, the *F*-measure is 80.54% when using the matrix **W**, instead of 72.33% when using M'. A comparison between **M** and M' has been made as well in order to provide evidence that M' provides better results than the original matrix. As a result, the *F*-measure is 65.54% when using **M**. Examples of facial image clusters produced by **M** can be seen in Fig. 16. The main drawback of using **M** is that of splitting clusters in many parts. As can be seen in Fig. 16, there are only few facial images of the same actor clustered together.

Table 1
Comparison of the *F*-measure between different clustering methods.

Clustering method	XM2VTS data set	Movie data set
N-cuts	100%	80.54%
k-means (for 7 clusters)	87.53%	67.53%
Fuzzy c-means (for 7 clusters)	92.64%	70.64%

Table 2
F-measure of the *N*-cut clustering method for various movies.

Movie used	*F*-measure	Optimal threshold	Actors really involved
American beauty	83.22%	0.05	2
Two weeks notice	80.54%	0.06	3
Platoon	74.63%	0.02	7
Jackie Brown	79.32%	0.06	2
Cold Mountain	82.01%	0.09	5
Analyze That	81.27%	0.06	4
Bring out the dead	89.42%	0.03	3
Dead poets society	98.30%	0.03	7
Indiana Jones & the last crusade	91.72%	0.05	4
Kids	92.74%	0.02	3
Lost highway	94%	0.02	10
Mission to Mars	83.74%	0.05	4
The pianist	92.37%	0.04	5
Pulp fiction	100%	0.06	2
I am Same	100%	0.02	3
Erin Brokovich	100%	0.06	4

Our method was compared to other clustering methods, namely k-means and fuzzy c-means [34] on the previously described data sets. In order to perform such a test, we assumed that each row of the matrix **W** forms an *N*-dimensional feature vector that can be used in the k-means and FCM algorithms ($N=350$ for the XM2VTS experiment and $N=941$ for the "Two weeks notice" one). The results are summarized in Table 1, where it can be clearly seen that the proposed algorithm outperforms the k-means and FCM algorithms.

Furthermore, in the k-means algorithm we need to predefine the cluster number, which is not the case for the *N*-cut algorithm.

After testing the proposed *N*-cut method on all movies, using the same procedure, we produced the results shown in Table 2. Approximately 1000 facial images per movie were used. The average *F*-measure obtained is around 80%, which indicates a very good performance in face clustering.

In order to compare our image similarity criterion to other ones, we have conducted experiments using a different MI-based similarity measure, the so called iso-contour mutual information (ISO-MI) [14]. The aforementioned technique is a geometric approach for determining the probability density function of the image intensities. The authors claim that this method overpasses problems in the calculation of the joint entropy arising from the geometrical transformation of the two images. In order to test our framework with this technique we have implemented this similarity criterion in our proposed approach.

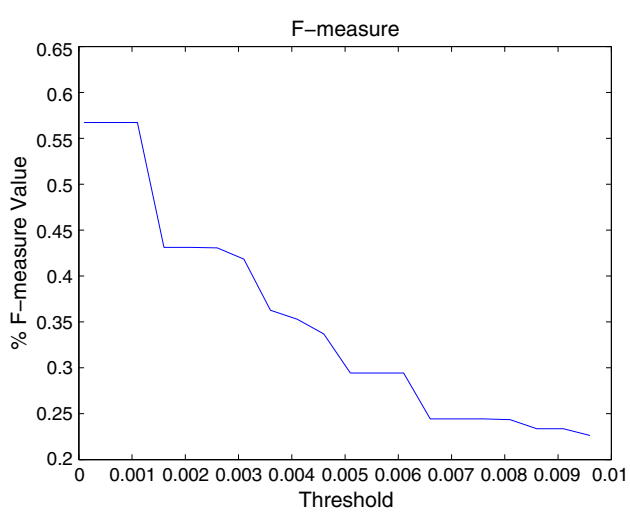

Fig. 17. *F*-measure for different clustering thresholds *T* for facial images from the movie "Two weeks notice" when using the ISO-MI as similarity criterion.

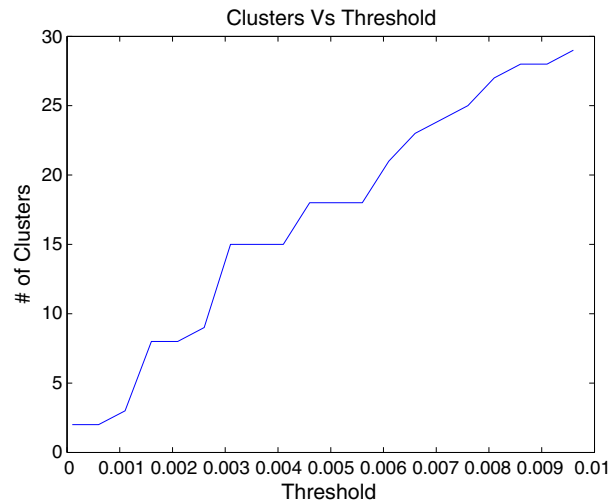

Fig. 18. Number of produced clusters for different clustering thresholds *T* for facial images of the movie "Two weeks notice" when using the ISO-MI as similarity criterion.

Results are depicted in Figs. 17 and 18, showing that the ISO-MI similarity measure produces much lower *F* values than the proposed hue/saturation MI measure shown in Fig. 10.

The main reason that this image similarity measure fails is the fact that it is based on image luminance. Another issue with the approach [14] is its computational complexity. It took more than 20 h to calculate the similarity matrix **M**, in contrast to the 6.5 h using the hue/saturation MI (Eqs. (5), (8), (9)) in the "Two weeks notice" experimentation video (i.e. 941 detected facial images).

Finally, when compared to other face clustering approaches, like the ones in [2] and [3], our method provides a fully automated framework, in contrast to these approaches. For instance, in [2], a training phase is performed to calculate the prior probabilities of the transformation parameters. In this phase, this algorithm needs manual eyes and mouth detection.

5. Conclusions

In this paper, we have described a face clustering algorithm that can be very helpful in semantic video analysis, e.g. in detecting dialogs or finding the principal actors in a movie. We have employed the mutual information in the HS domain as color facial image similarity measure. We demonstrated that the use of a global similarity measure of a facial image to a group of facial images outperforms the use of the simple image-to-image similarity. Our experimental results have an average *F*-measure of 80% in commercial movie clips, which shows that the proposed method can indeed be used in semantic movie analysis.

In future work, we shall concentrate our effort on improving the clustering procedure. More specifically, we shall investigate the automatic calculation of the clustering threshold value. Furthermore, we believe that the main drawback of the proposed method is its computation load due to the hue/saturation MI calculation. Therefore, we shall orient our work so as to minimize the computation effort needed to calculate the hue/saturation MI matrix **M**.

References

[1] N. Vretos, V. Solachidis, I. Pitas, A mutual information based algorithm for face clustering, Proc. of Int. Conf. on Multimedia and Expo (ICME 2006), 2006, (toronto Ontario, Canada, 9–12 July).
[2] A. Fitzgibbon, A. Zisserman, Joint manifold distance: a new approach to appearance based clustering, Computer Vision and Pattern Recognition, 2003, Proceedings. 2003 IEEE Computer Society Conference on, vol. 1, 2003, (pp. I–26 – I–33 vol.1).

[3] A. Fitzgibbon, A. Zisserman, On affine invariant clustering and automatic cast listing in movies, Proc. ECCV, 3, 2002, pp. 304–320.
[4] S. Foucher, L. Gagnon, M. CRIM, Automatic detection and clustering of actor faces based on spectral clustering techniques, Fourth Canadian Conference on Computer and Robot Vision, 2007. CRV'07, 2007, pp. 113–122.
[5] P.W. Pluim J, B.A. Maintz J, A. Viergever M, Mutual-information-based registration of medical images: a survey, IEEE Transactions on Medical Imaging 22 (8) (2003) 986–1004.
[6] Z. Cernekova, I. Pitas, C. Nikou, Information theory-based shot cut/fade detection and video summarization, IEEE Transactions on Circuits and Systems for Video Technology 16 (1) (January 2006) 82–91.
[7] E. Loutas, I. Pitas, C. Nikou, Probabilistic multiple face detection and tracking using entropy measures, IEEE Transactions on Circuits and Systems for Video Technology 14 (1) (2004) 128–135.
[8] Z. Liu, J. Yang, C. Liu, Extracting multiple features in the cid color space for face recognition, IEEE Transactions on Image Processing 19 (9) (2010) 2502–2509.
[9] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.
[10] H. Sossa, R. Horaud, G. LIFIA-IRIMAG, Model indexing: the graph-hashing approach, computer vision and pattern recognition, Proceedings CVPR'92, 1992 IEEE Computer Society Conference on (1992), 1992, pp. 811–814.
[11] S. Umeyama, An eigendecomposition approach to weighted graph matching problems, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (5) (1988) 695–703.
[12] M. Carcassoni, E.R. Hancock, Spectral correspondence for point pattern matching, Pattern Recognition 36 (1) (2003) 193–204.
[13] K. Messer, J. Matas, J. Kittler, J. Lüttin, G. Maitre, XM2VTSDB: the extended M2VTS database, Audio- and Video-based Biometric Person Authentication, AVBPA'99 Washington, D.C, March, 1999, pp. 72–77.
[14] A. Rajwade, A. Banerjee, A. Rangarajan, Probability density estimation using isocontours and isosurfaces: applications to information-theoretic image registration, IEEE Transactions on Pattern Analysis and Machine Intelligence (2008) 475–491.
[15] J. Pluim, J. Maintz, M. Viergever, f-information measures in medical image registration, IEEE Transactions on Medical Imaging 23 (12) (2004) 1508–1516.
[16] S. Kullback, R. Leibler, On information and sufficiency, Annals of Mathematical Statistics 22 (1) (1951) 79–86.
[17] J. Pluim, J. Maintz, M. Viergever, Image registration by maximization of combined mutual information and gradient information, IEEE Transactions on Medical Imaging 19 (8) (2000) 809–814.
[18] P. Thevenaz, M. Unser, Optimization of mutual information for multiresolution image registration, IEEE Transactions on Image Processing 9 (12) (2000) 2083–2099.
[19] J. Pluim, J. Maintz, M. Viergever, Interpolation artifacts in mutual information-based image registration, Computer Vision and Image Understanding 77 (2) (2000) 211–232.
[20] B. Likar, F. Pernus, Hierarchical approach to elastic registration based on mutual information, Image and Vision Computing 19 (1) (2001) 33–44.
[21] L. Paninski, Estimation of entropy and mutual information, Neural Computation 15 (6) (2003) 1191–1253.
[22] C. Studholme, Measures of 3D Medical Image Alignment, University of London, 1997.
[23] T. Cover, J. Thomas, Elements of Information Theory, Wiley, New York, 1991.
[24] J. Yang, A. Waibel, A real-time face tracker, IEEE Workshop on Applications of Computer Vision (1996) 142.
[25] H. Graf, T. Chen, E. Petajan, E. Cosatto, Locating faces and facial parts, Proceedings First International Workshop on Automatic Face and Gesture Recognition, 1995, pp. 41–46.
[26] K. Sobottka, I. Pitas, Looking for faces and facial features in color images, Pattern Recognition And Image Analysis C/c Of Raspoznavaniye Obrazov I Analiz Izobrazhenii, 7, 1997, pp. 124–137.
[27] S. Sural, G. Qian, S. Pramanik, Segmentation and histogram generation using the HSV color space for image retrieval, IEEE International Conference on Image Processing 2 (2002) 589–592.
[28] I. Cherif, V. Solachidis, I. Pitas, A tracking framework for accurate face localization, International Federation For Information Processing-publications-ifip, 217, 2006, pp. 385–389.
[29] F. Chung, Spectral Graph Theory, American Mathematical Society, , 1997.
[30] J. Cullum, R. Willoughby, Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Theory, Society for Industrial Mathematics, , 2002.
[31] B. Stein, S. zu Eissen, F. Wißbrock, On Cluster Validity and the Information Need of Users, Benalmádena, ACTA Press, Spain, September 2003, pp. 216–221.
[32] M. Krinidis, G. Stamou, H. Teutsch, S. Spors, N. Nikolaidis, R. Rabenstein, I. Pitas, An audio-visual database for evaluating person tracking algorithms, Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, Philadelphia, USA, 2005, pp. 452–455.
[33] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.
[34] R.L. Cannon, J.V. Dave, J.C. Bezdek, Efficient implementation of the fuzzy c-means clustering algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (2) (1986) 248–255.