A Robust Face Clustering Algorithm based on Mutual Information Using Tracking Information

N. Vretos, V. Solachidis and I. Pitas

Department of Informatics University of Thessaloniki,54124, Thessaloniki, Greece
phone: +30-2310996304, fax: +30-2310996304
email:vretos,vasilis,pitas@aiia.csd.auth.gr web: aiia.csd.auth.gr

Abstract. In this paper we create an algorithm in order to cluster faces. Our approach is based on the mutual information and more precisely its normalized version (NMI). In this paper we test two approaches one where we apply robust heuristics and another where we make use of a tracker in order to diminish dimensionality and augment accuracy of our results. It is a supervised clustering algorithm which is therefore used (fuzzy c-means) in order to gather same trajectories and same faces together.

1 Introduction

Face clustering is an important application for semantics extraction on video and can be used in a multitude of application in video processing. It can contribute in many ways, like determining the primordial actors or the creation of databases' references or dialog detection and many others. Until now some interesting algorithms have been proposed in [3]-[5], but most of them are based in calibrated face images from news or face recognition databases like [3]. Our approach exploits the capabilities of joint entropy and mutual information in order to classify face trajectories images exported from face tracker like the one proposed in [6].

Mutual information (MI) is a novel and useful tool in order to find similarities between information. More concretely, MI is defined as the information that is shared between two distributions. Until now, MI is much exploited in bioinformatics application and serves many purposes in that field from DNA sequences categorization [7] to classification of proteins [8]. In image processing MI is used, in many reprise, in image registration for medical images [8] and gives relatively good results. In this paper we will investigate the possibility of using this tool for classification of face images in a more realistic content such as movies, where difficulties arise from the fast variations of illumination, scale, pose etc.

The remainder of this paper is organized as follows: in Section 2 a mathematical presentation of the mutual information and its normalized version are presented, as well as the definition of heuristics and tracker's results integration. In Section 3 clustering algorithm is presented. In Section 4 we demonstrate results for a real movie case. Finally in Section 5 further work and conclusions are discussed.

2 Mutual Information for Face Clustering

Mutual information is defined as the information shared between two distribution. Let X and Y be two distributions. We define the joint entropy as:

$$H(X,Y) = -\sum(p(x,y)\log(p(x,y))) \tag{1}$$

where $p(x,y)$ the normalized (summed to one) probability density function of the common information of distribution X and Y. In the same way, we define the Shannon's entropy for X and Y as :

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

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

Therefore we can define the mutual information as:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{4}$$

where this give use the final equation of mutual information:

$$I(X;Y) = \sum_x \sum_y p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \tag{5}$$

$I(X;Y)$ is a quantity that measures the mutual dependence of two random variables. If we use a logarithm with base 2, then the measure is in bit. This quantity needs to be somehow normalized in order to create a uniform metric between different images and therefor be used as a similarity measure. For this reason, we use the normalized MI, which is defined as the quotient of the sum of two entropies with the joint entropy of those two distribution.

$$NMI(X;Y) = \frac{H(X) + H(Y)}{H(X,Y)} \tag{6}$$

Is is also useful to notice that:

$$NMI(X;Y) = \frac{H(X) + H(Y)}{H(X,Y)} \tag{7}$$

$$NMI(Y;X) = \frac{H(X) + H(Y)}{H(Y,X)} \tag{8}$$

But as we know from (1) :

$$H(X,Y) = H(Y,X) \tag{9}$$

So,

$$NMI(X;Y) = NMI(Y;X) \tag{10}$$

A very detailed explanation of how this normalizes the mutual information can be found in [9].

In our approach we use the intensity images and we define for every image the distribution density function as the histogram of the intensities of that image summed to one. In order to calculate the joint entropy between the two images we construct a 2D histogram of 256 bins which take in account the relative positions of intensities so that similarity occurs between two images, when same intensities are located in same spacial locations. Less literarily, the 2D joint histogram is calculated as follow: Let A and B be the first and the second image respectively of size $N_1 \times N_2$. And $i, j \in [0, 255]$ then:

$$Hist(i, j) = |\{(k, l) \in N_1 \times N_2 \mid A(k, l) = i \text{ and } B(k, l) = j\}| \tag{11}$$

where $|\cdot|$ denotes the cardinality of a set.

By defining the joint histogram that way, we have to admit that in order to calculate it, images have to be of same size.This means that one has to resize one image to the other's dimensions. In our approach, and in order to equalize big scaling interpolation issues, we define a mean bounding box which is calculated from all bounding boxes that the face detector provides to us. This approach shows better results than if we scale every pair of images forward the bigger or the smaller of them. So every image is scaled towards this mean bounding box before the mutual information calculation.

Another issue is the fact of anisotropic scaling. Detector's results are bounding boxes where typically width and height are not equal. In order to scale forward a mean bounding box problems arise when the two dimensions are not equal. To override this, we calculate the bigger dimension of the bounding box and then we are taking the square box that equals this dimension centered to the original's bounding box center. Less literally, let $B = \{x_1, y_1, x_2, y_2\}$ be a bounding box. We define the width as $x_2 - x_1$ and the height as $y_2 - y_1$. From the two dimensions we take the bigger one and stretch the other at that size. The resulting bounding box for B for width bigger than height (resp. height bigger than width), will be:

$$\begin{aligned} B_{new} &= \{x_1, y_1 - k, x_2, y_2 + k\} \\ (resp.\ B_{new} &= \{x_1 + k, y_1, x_2 - k, y_2\}) \end{aligned} \tag{12}$$

where k equals $\frac{(x_2-x_1)-(y_2-y_1)}{2}$.

We have noticed that problems arise from scaling issues that involves detectors inaccuracy. This means that if the face is not correctly detected and the face image contains a big amount of the background then scaling is mismatching the two face images and results are inaccurate. In order to override this bottleneck, another processing step is made, which is inspired from registration algorithms and aim in maximizing the accuracy of our results.

Once we have put the detector's outputs in the same scale we calculate the NMI for different frames of the target face image. We vary the bounding box's width and height from 80% to 120% of the initial mean bounding box, with a

step of 5%. The aforementioned values are calculated experimentally. In this way, we are trying to eliminate scaling problems due to detector's errors. In figure 1 one can see two images which show the aforementioned case. Finally, we take the maximum of the calculated NMIs between the two images.

As mentioned before, the movies' context is dominated from several difficulties in order to extract content information. The way our approach is using the mutual information is undertaking those problems. By using the scale variance within the detectors results and the point-to-point approach of the joint entropy we have succeeded to provide good results in a very complicated task. In [10] the problem is tackled based on a preprocessing of the image. Our approach is trying to avoid the preprocess and goes deeper in the mutual information properties to that end.

Fig. 1. In this image one can see that images are of different scales but faces are practically of same size .

2.1 Mutual Information Vectors

Our algorithm consists of creating a vector of MIs for every image. The dimension of that vector is equal to the size of the face detection results' data set. For every face image in the results set we calculate the NMI between this image and any other, and therefore we create a vector $\mathbf{v}$. All those vectors results in an $M \times M$ matrix (where M the cardinal of the set of all detections from a video sequence) where every row i of that matrix will be the NMI of the i-th detection with all other images.

$$S(i,j) = NMI(FaceImage_i, FaceImage_j) \tag{13}$$

It is obvious that the elements of the diagonal will have value one, which is the normalized mutual information of a face image with itself and also the matrix will be symmetric w.r.t the main diagonal. The diagonal property of the matrix is a forward effect of the MI symmetry shown in eq. (10). Those properties are

very helpful because they drastically intervene in the time complexity of the algorithm. By using those properties the time complexity is minimized by a multiplicative factor of 0.5 and an additive factor of -M. In figure 2 one can see the image of a matrix S for a 253 detections set. In this figure a test of consecutive appearances of two different actors is shown. One has to notice the square regions that appear in that image and that way we can understand that same persons appear. The thin lines that appears are in most cases detectors false results which are very different from the face pattern.

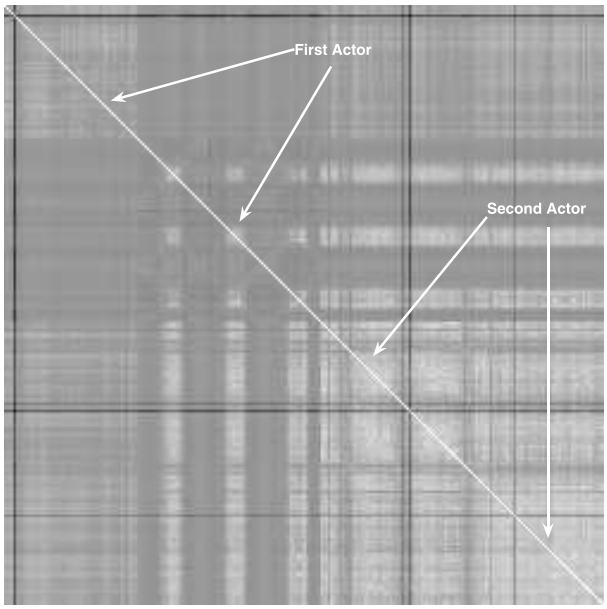

Fig. 2. Darker regions belongs to the first actor and clearer ones to the second actor. The video sequence has four consecutive shots in the order FA-FA-SA-SA where FA and SA first and second actor respectively.

2.2 Heuristics and Tracking Information integration

We have tested two methods in order to use the tracking information within our framework so as to generate better results than our previous work in [11]. The first approach is to heuristically modify the similarity matrix in a way that the face images within the same trajectories will have a mutual information value of one and also faces appearing in the same frame will have a mutual information of 0. Less literally, let S be the aforementioned similarity matrix extracted as

mentioned earlier. We create the new (robust) similarity matrix as follow:

$$S'(i,j) = \begin{cases} 1, & \text{if } (X_i, X_j) \ \textit{belong to the same tracking trajectory} \\ 0, & \text{if } (X_i, X_j) \ \textit{belong to the same frame} \\ S(i,j), & \text{if non of the above is true} \end{cases} \tag{14}$$

where X_i and X_j are two face images. The other approach consists of creating a similarity matrix from the tracking results. In this approach we calculate a statistical measure from the face's belonging to the same trajectory and therefore we create an $N \times N$ similarity matrix as follow:

$$ST(T_k, T_l) = f(B(k,l)) \tag{15}$$

where f is the statistics function (in our case the min,max,mean and median are used), and $B(k,l)$ is the submatrix of S' created from the cross entries of the trajectory k and l as follow:

$$B(k,l) = \{S'(i,j) |\ \ i \in k \ \ \textit{and} \ \ j \in l\} \tag{16}$$

The matrix ST is shown In figure 3.

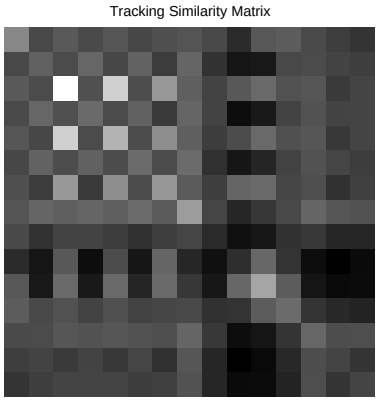

Fig. 3. Darker regions shows small amount of mutual information between tracker's results.

3 Fuzzy C-means Clustering

In order to cluster our results we use the fuzzy c-means algorithm. This method has been proven that in situation where we have a light mixture of classes'

elements, it performs better than the simple k-means algorithm. Other more sophisticated methods for clustering where abandoned because there where very consuming in time and did not give better results in this context. Therefore Fuzzy c-means (FCM) looked like the optimal solution in our case.

In order to use this algorithm we define every row of the similarity matrix S' as a different vector in an M-dimensional L^2-normed vector space over $\mathbb{R}$. In figure 4 and 5 one can see how those vectors are formed for two examples, one for 941-dimensional vectors with heuristics applied and another with 15-dimensional vectors from the tracker integration.

Therefore, we use the Euclidian distance to calculate distances between the vectors

$$dist(\mathbf{v_i}, \mathbf{v_j}) = \sqrt{\sum_{k=1}^{M} (v_{i_k} - v_{j_k})^2} \tag{17}$$

and by those means to calculate a predefined number of clusters' centers. A detailed implementation of the FCM algorithm can be found in [12].

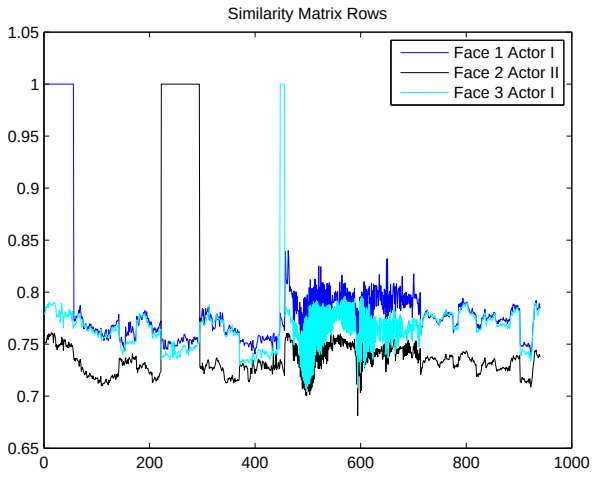

Fig. 4. 3 vectors which belong to 2 different actors from 3 distinct trajectories

We have seen that initialization has a significant role for FCM. So in order to provide better results the first centers are manually selected in a way that faces that corresponds in different actors constitutes a different initial center. A random selection of initial centers vary the results of a factor of 0.5% of false classification.

In the second approach the dimensionality of our vectors is drastically diminished due to the fact that M is now equal to the trajectories number and

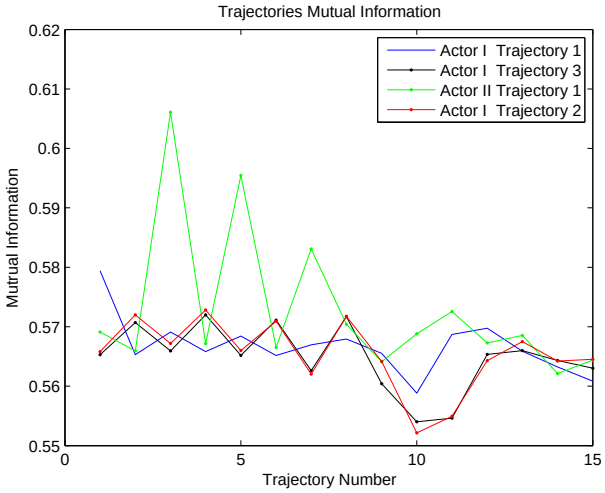

Fig. 5. 4 vectors which belong to 2 different actors from 4 distinct trajectories

not to the face images cardinality. So from the 941 face images which results to a 941-dimensional space we arrive at a level of 15 dimension.

4 Experiment Scenario and Results

In order to test our algorithm we have conducted the following experiment. From a movie called "Two weeks notice" we have extracted a set of 941 detections which belong to 3 different actors, and also they are tracked 15 times. First we execute the detection algorithm in every first frame of a shot or if for some reason the tracker stops, then we redetect for that frame and retrack until a shot boundary is encountered.

The frames where selected so that light conditions and actor's poses vary, and also we have selected pieces of the film from different scenes. In the detector's results set we end up with a variety of face images in many poses and light conditions. With this approach we ensure the robustness of our algorithm in those kind of attacks. In order to calculate the percentages of good and bad classifications we use a precision and recall like measure called F-measure [13]. The F-measure is calculated as follow: Let D represent a set and let $\mathcal{C} = C_1, ..., C_k$ be a clustering of D. Moreover, let $\mathcal{C}^* = C_1^*, ..., C_l^*$ design the human reference classification. Then the recall of cluster j with respect to class i, $rec(i, j)$ is defined as $\frac{|C_j \cap C_i^*|}{|C_i^*|}$. The precision of cluster j with respect to cluster i, $prec(i, j)$,

is defined as $\frac{|C_j \cap C_i^*|}{|C_j|}$. The F-measure combines both values as follows:

$$F_{i,j} = \frac{2}{\frac{1}{prec(i,j)} + \frac{1}{rec(i,j)}} \tag{18}$$

The overall F-Measure of clustering is given by:

$$F = \sum_{i=1}^{l} \frac{|C_i^*|}{|D|} \cdot \max_{j=1,..,k} \{F_{i,j}\} \tag{19}$$

We can easily note that a perfect fit between clustering and human reference leads to an F-measure score of 1. The F-measure is an external measure and thus it uses a human reference, i.e. it only shows how good the clustering is vis-a-vis to the human reference. The construction of the ground truth is mandatory for this process and unfortunately this kind of measure can not be used in real situations where human references are not available. Still, it is a very good measure for empirical evaluation of a new algorithm like the one proposed in this paper. In table 1 we can see the results of the F-measure for all the experiments

Method	F-Measure
FCM on MI	65.4%
FMC on Robust MI	67.6%
FMC on Tracker MI using Min	53.2%
FMC on Tracker MI using Max	**86.6%**
FMC on Tracker MI using Mean	75.0%
FMC on Tracker MI using Median	75.0%

Table 1. Results Table of F-Measure.

5 Conclusions And Future Work

We have developed a method for clustering face images within a very complex context such as movies. Results, as shown before, are rather promising for this difficult task if one considers the big variations that arise, w.r.t. light conditions, pose changes, emotions changes etc. in such a context. As face clustering has a lot of application in multimedia development, image processing and content based image retrieval applications (CBIR) we will investigate this problem further and we will concentrate our effort in the clustering process of the similarity matrix order to boost results.

The proposed method is a novel approach of the use that one can make of the mutual information in image analysis, and give good results in a hard task like the one we are solving. Exploration of the joint entropy and the mutual information on image data is shown to be a very good similarity criterion which can help in many other image processing application as well.

On the other hand, with our approach we minimize time complexity because of less preprocessing on the face images and the use of tracker information. This is an advantage for applications who needs fast clustering process, like interactive TV applications.

6 Acknowledgement

The work presented was developed within NM2 (New media for a New Millennium), a European Integrated Project (http://www.ist-nm2.org), funded under the European Commission IST FP6 program.

References

[1] S. Umeyama, "An eigendecomposition approach to weighted graph matching problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 5, pp. 695–703, 1988.

[2] Jianbo Shi and Jitendra Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[3] 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.

[4] 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.

[5] A.W. Fitzgibbon and A. Zisserman, "On affine invariant clustering and automatic cast listing in movies," *Proc. ECCV*, vol. 3, pp. 304–320, 2002.

[6] M. Krinidis, G. Stamou, H. Teutsch, S. Spors, N. Nikolaidis, R. Rabenstein, and I. Pitas, "An audio-visual database for evaluating person tracking algorithms," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, Philadelphia, USA, 18-23 March 2005, pp. 452–455.

[7] Xiaobo Zhou, Xiaodong Wang, Edward R. Dougherty, Daniel Russ, and Edward Suh, "Gene clustering based on clusterwide mutual information.," *Journal of Computational Biology*, vol. 11, no. 1, pp. 147–161, 2004.

[8] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, "Mutual-information-based registration of medical images: a survey," *Medical Imaging, IEEE Transactions on*, vol. 22, no. 8, pp. 986–1004, 2003.

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

[10] O. Arandjelovic and A. Zisserman, "Automatic face recognition for film character retrieval in feature-length films," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego*, 2005, pp. 860–867.

[11] N. Vretos, V. Solachidis, and I. Pitas, "A mutual information based algorithm for face clustering," in *Proc. of Int. Conf. on Multimedia and Expo (ICME 2006)*, 2006, Toronto Ontario, Canada, 9-12 July.

[12] R L Cannon, J V Dave, and J C Bezdek, "Efficient implementation of the fuzzy c-means clustering algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 2, pp. 248–255, 1986.

[13] B. Stein, S.M. zu Eissen, and F. Wißbrock, "On Cluster Validity and the Information Need of Users," *Benalmádena, Spain: ACTA Press, September*, pp. 216–221, 2003.