FACIAL IMAGE CLUSTERING IN 3D VIDEO USING CONSTRAINED NCUT

Georgios Orfanidis, Nikolaos Nikolaidis and Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki, GREECE

ABSTRACT

In this paper a novel variant of the Normalized Nut (N-Cut) clustering algorithm that incorporates imposed constraints is implemented and evaluated on facial image clustering for 3D video analysis. The clustering problem is seen as a graph cut problem through a similarity matrix representing the relation among the vertices, i.e. facial images in this work. Mutual Information is used as similarity metric, applied on the HSV color space of the original images. This work considers the incorporation of constraints either regarding similarity or dissimilarity derived from a priori available information in the clustering procedure and evaluates the performance increase by their use. Experiments are conducted on 3D videos where a priori information about the facial images exists.

Index Terms— Spectral clustering, Ncut, constraints, HSV color space, facial image clustering

1. INTRODUCTION

Clustering is one of the fundamental topics in computer science. It has been studied at a great degree but even at present day it is not considered solved. Clustering deals with dividing an existing set of object $\mathcal{P}$ (in our case, images) into a number of subsets (clusters) $\mathcal{C} = \{C_i | C_i \subseteq P\}$. Those subsets have to fulfill the following conditions: $\bigcup_{C_i \in \mathcal{C}} C_i = P$ and $\forall C_i, C_j, i \neq j \in \mathcal{C} : C_i \bigcap C_j = \oslash$. Essentially clustering assigns each sample of a data set to a non-overlapping set of groups, although there are exceptions to this rule like in the fuzzy c-means algorithm.

Clustering is related to other topics like classification and label propagation. Clustering is an unsupervised technique while classification belongs to the supervised techniques and label propagation to the semi-supervised ones. This paper deals with facial image clustering, where the goal is to separate face images into groups, for which within-cluster similarity is high whereas between-cluster similarity is smaller. This problem has been the subject of several previous works [1],[2]. Ideally each cluster should contain only instances of a single person and, if possible, all instances of that person. This means that if a person has many appearances all these appearances should be gathered in one cluster or, at least, the clusters should contain images from just one person. In other words, clusters should be at least homogeneous.

The rest of this work is organized as follows: Section 2 contains a short presentation of Mutual Information and its normalized version used in the proposed approach to evaluate similarity between images. In Section 3 spectral graph clustering using Normalised Cut is presented alongside with a limitation regarding its use. Section 4 deals with the imposition of constraints to Normalized Cut, Section 5 presents the results of the proposed method while Section 6 concludes the paper.

2. MUTUAL INFORMATION

Mutual Information (MI) is used in this work as the similarity measure among facial images, thus we will briefly present some related important definitions. Mutual Information is defined as the common information of two distributions. Entropy and joint entropy of two random variables X and Y are defined as:

$$H(X) \triangleq -\sum(p(x)log(p(x)))$$
$$H(X,Y) \triangleq -\sum(p(x,y)log(p(x,y)))$$

where $p(x)$ is the probability density function and $p(x,y)$ is the joint probability density function of random variables X and Y respectively. Mutual information is defined as:

$$\begin{aligned} I(X,Y) &\triangleq H(X) + H(Y) - H(X,Y) \\ &= -\sum(p(x,y)log(\frac{p(x,y)}{p(x)p(y)}) \end{aligned} \quad (1)$$

After applying normalization in (1) we get the normalized mutual information as in [3]:

$$D(X,Y) = \frac{H(X) + H(Y)}{2H(X,Y)}$$

In this work the similarity of images was evaluated on the HSV color space as it was shown to be more robust in illumination changes compared to RGB color space. More specifically we will follow the approach of [1] where only Hue (H) and Saturation (S) are used. According to this approach, 4D normalized MI is given by:

$$D(X,Y) = \frac{H(H_1) + H(S_1) + H(H_2) + H(S_2)}{2H(H_1, S_1, H_2, S_2)}, \quad (2)$$

where H_i is the Hue and S_i is the Saturation of each image respectively. Of the two Hue is considered more informative. For the 4D joint histogram an approach similar to [1] was used.

3. NORMALIZED CUT AND ITS LIMITATION

Normalized Cut (Ncut) [4] is a well-studied spectral graph clustering method applied in an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. It is defined primarily for bipartition, namely for splitting the graph into two parts, and it attempts to evaluate the cut that minimizes the edges (connections) between the two clusters and simultaneously maximizes the edges within the two clusters. An iterative approach can be applied in order to derive more than two clusters. While Ncut seems to work smoothly on bisection problems it reveals its limits when dealing with multi-class problems.

In order to use Ncut we define our face clustering problem as a graph cut problem. Each facial image is considered a vertex and the pairwise edge weight between each pair of vertices is given by the normalized MI (2) between them. We construct a similarity matrix $\mathbf{W}$ representing this graph in which the i-th row and j-th column element represents the edge weight between i-th and j-th vertex.

Having constructed the similarity matrix $\mathbf{W}$ we define the diagonal matrix $\mathbf{D}$ with elements $D_{ii} = \sum_j W_{ij}$ and the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Normalized cut minimizes the criterion $\text{argmin}_{y^T\mathbf{D1}=0} \frac{\mathbf{y}^T(\mathbf{D}-\mathbf{L})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}}$, where $\mathbf{1}$ is a vector of ones. It is proven in [4] that the eigenvector corresponding to the second smaller eigenvalue is the optimal graph cut for the given criterion.

The fact that Ncut functions on a bipartition basis is easily shown with toy examples as in Figure 1 where it fails to split in its first application the 4 clusters in a 3-1 manner and instead splits the first cluster in two parts. This rather unexpected behaviour reveals the impact of Ncut problem statement and the difficulties to expand it to multi-class problems. Ncut in its simplest form uses the line that separates positive from negative values and bipartition the clusters by grouping together all vertices having the corresponding element in the decision eigenvector positive or negative. As it is proven due to the nature of the relaxed solution being chosen some elements do not get a distinctive value but instead have a value near zero (ambiguous decision). To overcome this shortcoming we propose to search for the greater gap and not use a fixed threshold of zero. As it is shown in Figure 1, with this criterion Ncut can separate one cluster from the rest of the clusters. The same is true for the next eigenvectors. By this way the four clusters are easily identified and separated.

4. IMPOSING CONSTRAINTS IN NCUT

Ncut as defined in [4] does not take into consideration information that may be available during clustering. Such information could originate from various sources. In facial image clustering problems one could incorporate information from the face detector (two faces appearing into the same frame probably belong to different persons, with the exception of reflections on a mirror or window), tracker (a sequence of facial images belonging to the same tracking trajectory corresponds to the same person) or even an annotator (a human subject annotating facial images as belonging to the same or different persons). There have been some attempts to incorporate constraints into the general problem [5], [6], [7] so as to take advantage of the a priori information within spectral clustering but most works focus on bipartition problems (as Ncut itself). In our case those approaches proved inefficient and another solution was searched.

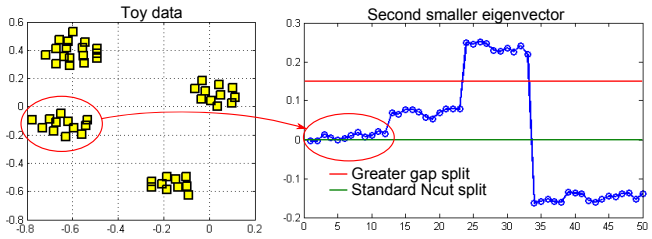


Fig. 1. Ncut unsuccessful separation in toy example

In order to modify the Ncut problem so as to include the constraints in the solution being searched, we apply some metric learning on the similarity matrix that corresponds to the graph. By this way we come up with a new similarity matrix $\mathbf{W_{new}}$ that incorporated the a priori information available and then we apply spectral clustering. Constraints are given as in other works that had dealt with the same problem namely, in pairs denoting similarity (dissimilarity) between two vertices. When two vertices are similar (dissimilar) this means that clusters in which they belong should obey to the same relation of similarity (dissimilarity). That is, if vertices C_k and C_l are considered similar (dissimilar) then with great certainty all vertices belonging to clusters C_k and C_l should be considered similar (dissimilar) as well.

To impose this, a notion that should be taken into consideration is the meaning of two clusters being similar or dissimilar. If, without loss of generality, we consider a similarity matrix that is block diagonal matrix (as a result of permutations for example), where the vertices of the first cluster appear first, then come the vertices of the second cluster etc, what we want to achieve is increase (decrease) the weights of the edges connecting the two clusters C_k and C_l. All other edges are not to be affected since we don't have information applicable to other clusters or vertices. Since we cannot just impose a value to those interconnecting blocks we need a way to determine: a) which vertices should be affected (belong to either of the two clusters) and b) by what intensity should they be affected.

Table 1 demonstrates the algorithm used for calculating

a new similarity matrix $\mathbf{W_{new}}$ by imposing the similarity(dissimilarity) constraints. The algorithm is quite self explanatory but we will discuss the general idea behind it. By definition, the similarity matrix is a matrix revealing the similarities among the various vertices of the graph. In our case, each vertex represents a facial image so it reveals the similarity among facial images. Obviously, it is expected that within a cluster all vertices would have high similarity scores while vertices belonging to different clusters would have lower scores. Having this in mind, imposing a similarity (dissimilarity) constraint onto 2 vertices belonging to different clusters would mean to increase (lower) the inter-cluster similarities. Similarities within each cluster should not be affected as the imposed constraint does not affect the relations within each cluster. The application of this idea is demonstrated in Figure 3 where a toy example of 40 samples is used.

We have chosen to demonstrate the algorithm on a toy example for the sake of visual demonstration of the changes occurring to the similarity matrix. One final point that should be pointed out is that the algorithm presented in Table 1 shows that the more dissimilar the two vertices i and j the higher the edge weights of matrix $\mathbf{M}$. This clearly represents the desired behaviour for similarity constraints since we virtually want to apply changes on the weights of edges among clusters that the similarity matrix does not represent sufficiently as similar whereas based on but considering the constraints they should be similar. At the same time, we wish to keep as unchanged as possible the edge weights that should be similar (according to the imposed constraints) and are considered indeed similar (according to the similarity matrix). If we consider the case where the two vertices belong to the same cluster then, with great a deal of certainty, we can assume that they share similarities with all other vertices that do not differ significantly. In that case the non-zero elements of the vectors $\mathbf{v} - \mathbf{u_i}$ and $\mathbf{v} - \mathbf{u_j}$ would have negligible value and thus matrix $\mathbf{M}$ would have very low maximum weight and therefore, the new similarity matrix ($\mathbf{W_{new}}$) would not differ substantially from the original one ($\mathbf{W}$). In the opposite case where the two vertices i and j are quite dissimilar but according to constraints belongs to the same cluster then the non-zero elements of the vectors $\mathbf{v} - \mathbf{u_i}$ and $\mathbf{v} - \mathbf{u_j}$ would have significant values and matrix $\mathbf{M}$ would also have a significantly large maximum value but only in the edges connecting the two clusters of vertices i and j. The above refers to the case of a similarity constraint but the general idea could be applied (inverted) to a dissimilarity constraint among a pair of vertices.

Table 1. Algorithm for introducing constraints to the Ncut algorithm

Preliminary: Similarity matrix is $\mathbf{W}$ and constraints are given into pairs between vertices i and j	
Repeat	
Step #1	Define vectors $\mathbf{u_i}$ and $\mathbf{u_j}$ corresponding to i-th, j-th columns of $\mathbf{W}$
Step 2	Calculate vector $\mathbf{v} = \mathbf{max}(\mathbf{u_i}, \mathbf{u_j})$
Step 3	Calculate matrix $\mathbf{M} = (\mathbf{v} - \mathbf{u_i})(\mathbf{v} - \mathbf{u_j})^\mathbf{T} + (\mathbf{v} - \mathbf{u_j})(\mathbf{v} - \mathbf{u_i})^\mathbf{T}$
Step 4	Add or subtract matrix $\mathbf{M}$ from matrix $\mathbf{W}$ · in case of similarity contraint → $\mathbf{W_{new}} = \mathbf{W} + \alpha\mathbf{M}$ · in case of dissimilarity contraint → $\mathbf{W_{new}} = \mathbf{W} - \alpha\mathbf{M}$ where α is a value usually set to $\alpha = max(\mathbf{v})$
While	there are still constraints
	Apply Ncut algorithm to the new similarity matrix $\mathbf{W_{new}}$

5. EXPERIMENTAL RESULTS

We have applied our algorithm to various short 3D videos so as to estimate its performance. In order to measure this performance we used the F-measure [8], also known as F_1 and F_{score}, that takes into consideration both precision p and recall r. F-measure takes values in the range [0,1], with 0 being the worst and 1 being the perfect score, and it is evaluated as a weighted average of precision and recall. Given a set P, a certain clustering $\mathcal{C} = \{C_1, ..., C_K\}$ and the Ground Truth clustering $\mathcal{C}^* = \{C_1^*, ..., C_k^*\}$ then the recall of cluster j with respect to cluster i, $r(i,j)$ is defined as $\frac{|C_j \cap C_i^*|}{|C_i^*|}$ and the precision of cluster j with respect to cluster i, $p(i,j)$ is defined as $\frac{|C_j \cap C_i^*|}{|C_j^*|}$. Consequently the F-measure for the two clusters is defined as $F_{i,j} = 2\frac{p(i,j)r(i,j)}{p(i,j)+r(i,j)}$, where $|.|$ is the cardinality of the set. Combining all F-measures we obtain the overall F-measure: $F = \sum_{i=1}^{L} \frac{|C_j^*|}{|P|} \max_{j=1,...,k}\{F_{i,j}\}$.

In order to evaluate the performance of the proposed method we tested it against the standard Ncut method and Ncut using greater gap threshold method. Results are shown in Table 2 where GG Ncut refers to Greater gap Ncut and Con Ncut to Ncut with constraints (proposed method) respectively. In order to evaluate the effect of the number of constraints on the method performance we imposed random $p\%$ constraints for each movie for $p = 2\%$, $p = 5\%$ and $p = 10\%$. Experiments were repeated five times to obtain average results and were conducted in three ways: to left channel, to right channel and to both channels of the stereoscopic video. No other constraint besides the previously mentioned was used. Since the experiments were performed on stereoscopic videos two facial images, for the left and the right channel, were considered per frame. Although these two images are known to correspond to the same person and thus to the same character, this information was not taken into account. In other words only few constraints of this type (left and right facial images correspond to the same cluster), namely those in the

3D film clustering performance			
movie #1			
method	channel		
	left	right	both
Ncut	43.12%	43.17%	42.50%
GB Ncut	45.25%	46.02%	44.79%
Con NCut p=2%	48.39%	49.71%	48.06%
Con NCut p=5%	58.65%	61.51%	62.89%
Con NCut p=10%	76.97%	77.67%	75.43%
movie #2			
method	channel		
	left	right	both
Ncut	59.12%	59.03%	57.52%
GB Ncut	63.32%	59.31%	56.81%
Con NCut p=2%	67.76%	66.39%	64.32%
Con NCut p=5%	76.78%	77.26%	76.28%
Con NCut p=10%	82.39%	82.06%	80.22%
movie #3			
method	channel		
	left	right	both
Ncut	44.33%	42.53%	43.15%
GB Ncut	44.82%	43.89%	43.70%
Con NCut p=2%	45.71%	48.03%	52.52%
Con NCut p=5%	62.63%	61.69%	75.14%
Con NCut p=10%	80.00%	80.90%	87.39%

Table 2. Performance on various 3D short films

Class cardinality of each movie			
movie	#1	#2	#3
Class cardinality	14	9	4

Table 3. Number of classes for the test movies

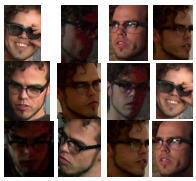
$p\%$ randomly chosen constraints each time, were taken into account. By not exploiting all constraints available in case of stereo channel we are not being fair against the stereo channel but the purpose of the experiments was to demonstrate the influence of constraints regardless the nature of their origin. Performance on stereo channel is expected to be higher if all constraints available were imposed. The percentage of constraints plays an important role in the performance as expected. On the other hand the gain in performance was in general much higher than the percentage $p\%$ itself, a fact that reveals good generalization of the constraint to other similar (or dissimilar) samples.

It must be noted that the F-measure requires the ground truth of the samples, in our case the class in which each facial image belongs.

The facial images used were taken from short films and every dataset comprised of facial images in various poses and illumination conditions. Faces were manually selected and then tracked automatically over frames, which accounts for possible errors that could have occurred. Some samples are shown in Figure 2 where the variations within two classes are presented. So we expect different clusters to be created that are separate apart but belong to the same class. In these cases the constraints come handy. The possible variations among facial images within each class include scale, illumination, pose, occlusion and expressions which creates a large space for samples to be spanned. As it is expected the images that comprise such classes are quite difficult to be successfully assigned to just one cluster by a clustering method. However, by imposing constraints the algorithm can be forced to join that were erroneously split. Up to a point the method used in this work can ameliorate the difficulties imposed by to some of these variations. For example it can deal quite well with pose variations if other conditions are kept fixed, but it could not deal with all of them. Another factor that increases the difficulty of clustering is the number of the classes. Other works that introduce constraints in Ncut [5], [6] consider mainly the bipartition case which may be valid for segmentation applications but it's totally inadequate for facial image clustering. In contrast, our work focuses on multi-class problems. The class cardinality of each movie is shown in Table 3.

(a) Facial images examples

(b) Class A

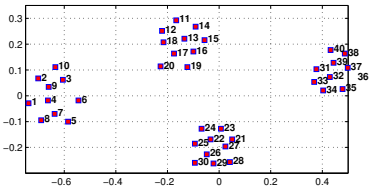


(c) Class B

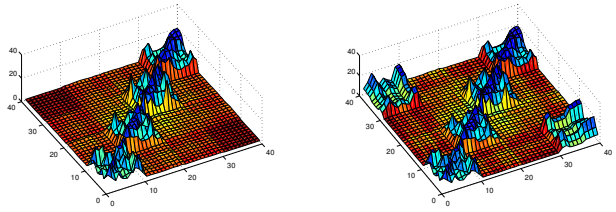
Fig. 2. Facial image samples from a film and variation within two classes of samples

6. CONCLUSION

In this work we presented a method for exploiting a priori information that might exist, such as constraints, for improving clustering performance. Such information could originate from various sources and in this work we did not examine the origin so as to show the general applicability of the proposed algorithm. The proposed method improves significantly the performance especially in cases where class oversplitting into many clusters occurs. In the future we intend to use the extra information encapsulated in a 3D video like the similarity of facial images belonging to the same person on the two channels, or the information that could be derived by the face detector or tracker.



(a) Toy example



(b) Original Similarity matrix (c) New Similarity matrix

Fig. 3. Toy example demonstrating the effect of imposing one similarity constraint between clusters #1 (vertex #4) and #4 (vertex #39)

7. ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 287674 (3DTVS). This publication reflects only the authors views. The European Union is not liable for any use that may be made of the information contained therein.

8. REFERENCES

[1] N. Vretos, Solachildis V., , and Pitas I., "A mutual information based face clustering algorithm for movie content analysis," *Image and Vision Computing*, vol. 29, pp. 693–705, 2011.

[2] Foucher Samuel and Langis Gagnon, "Automatic detection and clustering of actor faces based on spectral clustering techniques," in *Fourth Canadian Conference on Computer and Robot Vision*. IEEE, 2007.

[3] J. Pluim, J. Maintz, and M. Viergever, "Image registration by maximization of combined mutual information and gradient information," *IEEE Transactions on Medical Imaging*, vol. 19, pp. 809–814, August 2000.

[4] Shi Jianbo and Jitendra Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, August 2000.

[5] Xu Linl, Wenye Li, and Dale Schuurmans, "Fast normalized cut with linear constraints," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.

[6] Stella X. Yu and Jianbo Shi, "Grouping with bias," Neural Information Processing Systems (NIPS), 2001.

[7] Ji Xiang and Wei Xu, "Document clustering with prior knowledge," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006.

[8] Stein Benno, S. Meyer zu Eissen, and Frank Wissbrock, "On cluster validity and the information need of users," in *Artificial Intelligence and Applications*. ACTA Press, 2003.