# Stereo facial image clustering using double spectral analysis

Georgios Orfanidis,[1,*] Nikos Nikolaidis, [1], Anastasios Tefas, [1] and Ioannis Pitas [1]
[1]Department of Informatics
Aristotle University of Thessaloniki, GREECE
*eypros@aiia.csd.auth.gr

***Abstract*—In this work we proposed a new variant of spectral clustering using double spectral analysis which proved to be able to achieve better clustering results. The present work focuses on the special case of 3D videos and the implication of their use. Extended experiments have been conducted in three 3D full feature commercial films which revealed the power of stereo face clustering in comparison with single channel face clustering.**

***Index Terms*—3D, stereo facial image clustering, Spectral clustering, Ncut, HSV color space, facial image clustering**

## I. 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 objects $\mathcal{P}$ (in our case, images) into a number of subsets (clusters) $\mathcal{C} = \{C_i | C_i \subseteq \mathcal{P}\}$. 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 cluster, although there are exceptions to this rule like in the fuzzy c-means algorithm.

This paper deals with facial image clustering applied on 3D commercial feature films, where the goal is to separate face images into groups, for which within-cluster similarity is high whereas between-cluster similarity is smaller. There is some work with single channel feature films like [1] but similar work on 3D films is quite limited. Also, in [1] these films there is limited variance in optical effect, illumination, or special features of the images in the used test films. The film used in [1] has just 1 shots with consistent clothing, the film focuses on character development rather than visual effect etc. There are also various works with facial image clustering not deriving from videos like [2], [3] but focusing on still images. in this case the requirements are different. On the other hand, even in mono videos, often the dataset derives from much shorter videos as in [4] or is confined as in [5]. Spectral clustering has also been used in previous works [6],[7].

The rest of this work is organized as follows: Section 2 contains a short presentation of Mutual Information and its normalized version used in this work as similarity measure among images. Section 3 provides the problem statement and prior work on the field while Section 4 introduces the double spectral clustering and the improvements proposed. Section 5 contains the experimental results and finally Section 6 concludes the paper.

## II. Mutual Information

Mutual Information (MI) is used in this work as the similarity measure between facial images, thus we will briefly present some related important definitions. the interested reader may refer to [6] for details on MI.

Mutual Information is defined as the common information of two distributions. Entropy of a random variable $X$ is defined as:

$$H(X) \triangleq -\sum(p(x)log(p(x))) \tag{1}$$

while joint entropy of two random variables $X$, $Y$ is defined as:

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

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. The normalized mutual information used is:

$$D(X,Y) = \frac{H(X) + H(Y)}{2H(X,Y)} \tag{3}$$

as in [8]. In this work the HSV color space of images was used for calculating similarity as it has been proved to be more robust in illumination changes compared to RGB color space. More specifically we will follow the approach of [6] where only Hue (H) and Saturation (S) are used. According to this approach, a 4D normalized MI is used:

$$D(X,Y) = \frac{H(H_1) + H(S_1) + H(H_2) + H(S_2)}{2H(H_1, S_1, H_2, S_2)} \tag{4}$$

where $H_i$ is the Hue and $S_i$ is the Saturation of each image respectively. For the 4D joint histogram an approach similar to [6] was used.

## III. Problem statement and prior work

In order to use spectral clustering techniques we define our face clustering problem as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ cut problem. We consider each facial image to be a graph vertex and then we construct a similarity matrix $\mathbf{W}$ in which each element $w_{ij}$ represents the edge that connects the two images (vertices), which has a weight representing the similarity of the corresponding image pair and has an assigned value of $1/d_{ij}$.

We can then use different spectral approaches to solve the problem. Crucial to all approaches is the definition of the Laplacian matrix $\mathbf{L}$. The normalized Laplacian is defined as:

$$\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} \tag{5}$$

where $\mathbf{D}$ is the Degree matrix. The normalized Laplacian $\mathbf{L}$ has been used in our case since it has been proven to have greater merits than the unnormalized one [9].

Having defined the Laplacian matrix $\mathbf{L}$ different approaches for clustering can be used. One possible solution could be the use of Normalized Cuts (Ncut) [10] described in subsection

III-A while another solution could be the approach introduced in [11] and summarized in subsection III-B below.

### A. Normalized cut

Ncut is defined primarily for bipartition, namely for splitting the graph into two parts, and it attempts to find the cut that minimizes the edge weights (connections) between the two clusters while simultaneously maximizing the edge weights within the two clusters.

The problem the Ncut algorithm tries to solve is defined as $\text{argmin}_{\mathbf{y}} \frac{\mathbf{y}^T(\mathbf{D}-\mathbf{W})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}}$ subject to $\mathbf{y}^T\mathbf{D}\mathbf{1} = 0$ which uses the matrices defined in section III. The Ncut problem is usually solved by relaxing the solution using eigen-analysis [10].It is proven in [10] that the eigenvector corresponding to the second smallest eigenvalue is the optimal graph cut for the given criterion. More eigenvectors or an iterative eigen-analysis can be applied in order to derive more than two clusters.

### B. Spectral clustering

While Ncut solves a binary problem, the spectral clustering approach proposed in [11] solves a multiclass clustering problem. This solution also uses the Laplacian matrix $\mathbf{L}$ defined in (5). The solution applies eigen-analysis in $\mathbf{L}$ and uses the first $k$ eigenvectors $\mathbf{u}_2, ..., \mathbf{u}_{k+1}$ excluding the first one. A new matrix containing those eigenvectors is defined as $\mathbf{U} \in \mathbb{R}^{N\times k}$ ($N$ being the number of images). New samples are defined as the rows of this matrix after being normalized to norm 1. The final step is to use $k$-means [12] to these new samples.

## IV. Introducing double spectral analysis

The two approaches briefly mentioned in sections III-A and III-B although have proved quite efficient in various cases, have some drawbacks. In this Section, we list these drawbacks and also propose ways to overcome them while preserving their merits.

At first, Ncut is designed to solved two-class problems which can have some implications when applied to multi-class problems. One simple example is demonstrated in figure 1(a). In this example a simple similarity matrix $\mathbf{W}_{sample}$ is used, created by inverting the Euclidean distance of each pair of points $\mathbf{x}_{ij}$.

Assume that we cluster the samples using the Ncut algorithm and matrix $\mathbf{W}_{sample}$. We plot the elements of the second smallest eigenvector in Figure 1(b) which is the optimal solution for the Ncut problem according to [10] and also choose to use the simplest threshold, namely zero. While there are 4 easily distinguishable classes, shown in Figure 1(a), Ncut fails to separate correctly the classes as it is obvious in Figure 1(b). More specifically, class #1 is split in two by using this threshold. By observing the elements of the eigenvector corresponding to each sample we see that all elements corresponding to samples of the same class obtain similar values. That observation led us to use a more flexible threshold, the greatest gap threshold (GG threshold) as explained in section IV-B.

The main drawback of the spectral clustering approach [11] is the use of $k$-means as the main clustering algorithm. It is well known that $k$-means has various limitations; such as the assumption of spherical clusters, the effect of initialization on its performance and the possible convergence to a local minimum. Thus, we opted for a more efficient clustering algorithm. Since Ncut does not assume the above we opt to use it instead of $k$-means, as described in the next Section.

### A. Double spectral clustering

The proposed approach is similar to the one in [11]. Assume that we have a dataset with $N$ facial images. We then calculate the similarity matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$ as described in Section II, III and the normalized Laplacian matrix $\mathbf{L} \in \mathbb{R}^{N\times N}$ in (5).

We apply eigen-analysis to $\mathbf{L}$ and create a new matrix $\mathbf{U}$ containing the $k$ first eigenvectors of $\mathbf{L}$ as columns. The eigenvectors are ordered using their corresponding eigenvalue in ascending order. Furthermore, the first eigenvector corresponding to the trivial solution is not taken into consideration. Thus, $\mathbf{U} = \{\mathbf{u}_2, ..., \mathbf{u}_{k+1}\}$.

We create new samples $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_N\}$ that correspond to the rows of matrix $\mathbf{U}$. Those new samples have dimension $k$ which equals the number of eigenvectors being used. It must be noted that we do not normalize the samples $\mathbf{y}_i$ as in [11], in order to keep as much discriminality as possible. Normalization using $L_2$ norm would mean projecting all samples to the unitary circle which obviously reduces the distance among samples.

The general idea is to create new samples that benefit from the eigen-analysis of the Laplacian matrix $\mathbf{L}$ which tends to gather together samples with high similarity score in $\mathbf{W}$ and to distanciate samples with low similarity score in $\mathbf{W}$.

From this point we create a new similarity matrix $\mathbf{W}'$ considering as initial data set the $\mathbf{Y} \in \mathbb{R}^{N\times k}$ and using the inverse of the Euclidean distance of similarity score:

$$[\mathbf{W}']_{i,j} = \frac{1}{(\mathbf{y}_i - \mathbf{y}_j)^T(\mathbf{y}_i - \mathbf{y}_j)} \tag{6}$$

Then, we proceed by creating the correspoding Laplacian matrix $\mathbf{L}'_{sim}$ and apply the Ncut algorithm in order to split the dataset into 2 subclusters. In order to get the desired number of clusters we recursively apply the Ncut algorithm to those subclusters until this number has been reached.

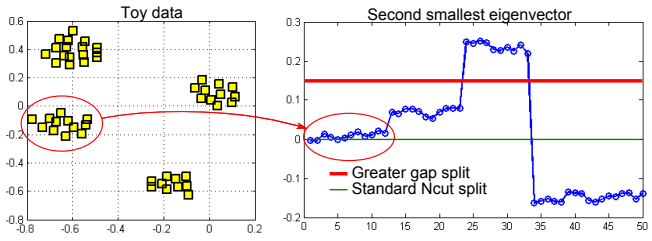
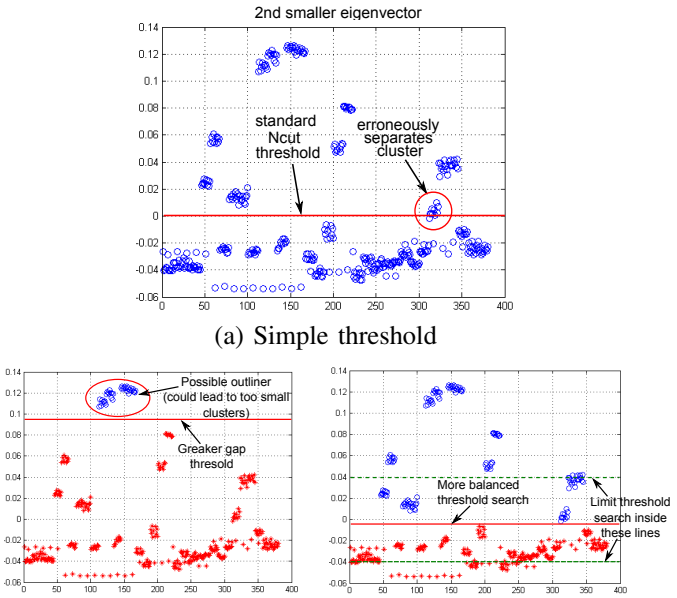



Fig. 1. Ncut unsuccessful clustering in toy example

### B. Greater gap threshold

As it has been previously mentioned and also shown in Figure 1 the simple thresholding approach that uses a threshold equal to zero on the eigenvector elements often fails when used with multi-class problems. For this reason we propose a new approach which uses a more flexible threshold that has been proven to be more reliable. Figure 2 plots the elements of the eigenvector of a real problem that corresponds to the second smallest eigenvalue and their relative position with respect to the zero axis. One possible problem with a fixed zero threshold has already been mentioned and concerns the separation of clusters which get values around zero, as the one




(a) Simple threshold

(b) Overall greatest gap threshold

(c) Limited greatest gap threshold

Fig. 2. The effect of using different thresholds

shown in Figure 2(a) inside the red circle. Those clusters could have samples which correspond to very similar elements of the eigenvectors (and thus they should not be split) but the a fixed zero threshold does not take into account this similarity.

Thus, instead of using a fixed threshold, we first sort the elements of the second eigenvector and then search for the greatest difference between two consecutive elements. Then the threshold is set as the mean value of these two consecutive elements having the greatest gap. Thus we get a different threshold in each application of the Ncut algorithm.

One problem with this approach is that it tends to first separate the outliers from the other samples (Figure 2(b)) which may not be an optimal approach. For this reason we restrict the threshold search inside some limits as shown in Figure 2(c). The restriction limits are usually defined with respect to the greatest value $v$ occurring in each eigenvector, that is as a proportion of this greatest value $v$: $Limits : \pm av,\ a \in (0, 1]$.

## V. Experiments

### A. Mono and stereo dataset creation

We applied our proposed method in three full length 3D feature films of different duration, size of cast and genre. For the creation of the facial image dataset, face detection and tracking was applied. Two different approaches were tested, in an effort to highlight the advantage of using stereo data. In the first approach, the face detection [13] is performed every $n$ frames in one channel (video) of the stereoscopic video, followed by a face tracker [14], [15] at the same channel. In the second approach, face detection [13] was applied on both channels, mismatches between the two channels were rejected and a stereo tracking algorithm [15] was applied in both channels. By using the above approaches we end up with a number of facial trajectories, namely series of consecutive facial images.

As can be observed in Table I the trajectories generated by the stereo face detectors and trackers are fewer in number and longer (in number of frames). This has the advantage of fewer facial images to be clustered and better representation of the actor appearances. Another observation that can be made in Table I is that the number of clusters (that equals the cast size) is quite high. The cardinality of actor clusters varies significantly with some actors having many facial images while others have very few appearances.

Usually, each tracking trajectory is represented by one (usually detected) facial image. Another approach for dataset creation examined was the inclusion of multiple facial images per trajectory. We used all detected images as well as each middle image between two sequential detections belonging to the same trajectory. For 2 trajectories represented by 2 sequences of $k$ and $m$ images respectively, $T_k$ and $T_m$ we calculate the similarity between them as:

$$\text{Similarity Score} = \max_{ij} MI(x_i, x_j) \quad x_i \in T_k, x_j \in T_m$$

where $x_i$ is an image belonging to the sequence $T_k$. The extreme values for the size of $T_i$ is 1 (as the usual case) to the whole set of images belonging to each trajectory. By this way a robustness in variation in pose, illumination etc is applied. It can be seen that with multiple images the number of images is increased but the similarity matrix used in clustering is of the same size as before (having the size of the number of trajectories) because only one score per trajectory is finally used. Experiments have been conducted using this new approach and revealed an increase in performance.

### B. Experimental results

Experiments were conducted in three 3D feature films with varying duration, size of cast, special effect etc. Some characteristics of each film are shown in Table I where it can be seen that stereo dataset are fewer in number and bigger (in number of frames). This has the advantage of fewer facial images to be clustered and also larger trajectories which signifies better representation of the films. Experimental results confirm this also in the performance. In Table I it is obvious that the number of clusters (that equals the cast size) is quite high while another fact is the unbalanced classes; with some persons having many images and also some persons having very few appearances.

Results have been accumulated in Table II where it can be observed that there is a significant improvement over Ncut by using both double spectral clustering and double spectral clustering with constrained greatest gap. Of the two latter methods best results were achieved using the double spectral clustering in combination with greatest gap in all 3 films. The use of stereo video also improves the performance and provides an extra way to further improve the results. There is a significant advantage in using both channels in comparison with a single channel.

For the evaluation of the clustering a variation of $F$-measure was used. Since the number of clusters is quite high the standard $F$-measure tends to punish the splitting of classes into more clusters. To evaluate more the purity of clusters we use a different approach: we oversplit the set to more than the clusters needed and then merged them in a way a user would merge them; that is by merging all clusters having the majority of their elements derive from the same class.

TABLE I
VARIOUS CHARACTERISTICS OF THE THREE FEATURE FILMS

| Size of cast | # of frames | # of trajectories | |
|---|---|---|---|
| | | in stereo | in single channel |
| Feature film #1 | | | |
| 27 | 181,763 | 1246 | 1545 |
| Feature film #2 | | | |
| 47 | 150,361 | 1222 | 1559 |
| Feature film #3 | | | |
| 58 | 196,224 | 1726 | 2238 |

## VI. CONCLUSION

In this work we proposed a new variant of spectral clustering using double spectral analysis which proved to be able to achieve better clustering results. We also focus mainly on 3D films and the special issues regarding their use, like the possible improvement by the use of stereo video compared to mono ones, but also in other aspects of films like the use of multiple images per trajectory for further improving the results. Finally a more flexible threshold was introduced that also proved more efficient. Further work will include more extensive research on the use of 3D films for clustering purposes.

## REFERENCES


[1] Vineet Gandhi and Remi Ronfard, "Detecting and naming actors in movies using generative appearance models.," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[2] WT Chu, YL Lee, and JY Yu, "Visual language model for face clustering in consumer photos.," *Proceedings of the 17th ACM international conference on Multimedia*, 2009.

[3] Edoardo Ardizzone, Marco La Cascia, and Filippo Vella, "Mean shift clustering for personal photo album organization.," *IEEE International Conference on Image Processing*, 2008.

[4] S. Schwab, T. Chateau, C. Blanc, and L. Trassoudaine, "A multi-cue spatio-temporal framework for automatic frontal face clustering in video sequences.," *EURASIP Journal on Image and Video Processing*, vol. 1, pp. 1–12, 2013.

[5] Chen Guangliang and Gilad Lerman, "Spectral curvature clustering (scc).," *International Journal of Computer Vision*, vol. 81, pp. 317–330, 2009.

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

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

[8] 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.

[9] Ulrike Von Luxburg, "A tutorial on spectral clustering.," *Statistics and computing*, vol. 17, pp. 395–416, 2007.

[10] 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.

[11] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm.," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[12] Stuart Lloyd, "Least squares quantization in pcm.," *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, 1982.

[13] GN Stamou, M Krinidis, N Nikolaidis, and I Pitas, "A monocular system for automatic face detection and tracking," *Visual Communications and Image Processing (VCIP)*, vol. 5960, pp. 794–802, 2005.

[14] Haris Baltzakis, Antonis Argyros, Manolis Lourakis, and Panos Trahanias, "Tracking of human hands and faces through probabilistic fusion of multiple visual cues," *Computer Vision Systems*, pp. 33–42, 2008.

[15] O. Zoidi, N. Nikolaidis, and I. Pitas, "Appearance based object tracking in stereo sequences.," *In 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.


TABLE II
RESULTS ON THE 3 FEATURE FILMS FOR VARIANTS OF NCUT AND IMAGE/TRAJECTORY SIMILARITY

| Feature film #1 | | | | |
|---|---|---|---|---|
| | Stereo Left | | Mono Left | |
| | # of images | | # of images | |
| | 1 | k | 1 | k |
| Ncut | 0.3030 | 0.3984 | 0.3690 | 0.4096 |
| Double spectral | 0.4341 | 0.5234 | 0.5073 | 0.5234 |
| GG Double spectral | 0.4658 | 0.5419 | 0.5051 | **0.5446** |
| Feature film #2 | | | | |
| | Stereo Left | | Mono Left | |
| | # of images | | # of images | |
| | 1 | k | 1 | k |
| Ncut | 0.2272 | 0.3274 | 0.2235 | 0.3246 |
| Double spectral | 0.2989 | 0.3574 | 0.3379 | 0.3785 |
| GG Double spectral | 0.3804 | 0.4576 | 0.4111 | **0.4626** |
| Feature film #3 | | | | |
| | Stereo Left | | Mono Left | |
| | # of images | | # of images | |
| | 1 | k | 1 | k |
| Ncut | 0.1554 | 0.2183 | 0.1600 | 0.2231 |
| Double spectral | 0.2986 | 0.3490 | 0.3409 | 0.3305 |
| GG Double spectral | 0.3179 | 0.3607 | 0.3349 | **0.3710** |