# Facial Image Clustering in Stereo Videos Using Local Binary Patterns and Double Spectral Analysis

Georgios Orfanidis, Anastasios Tefas,
Nikos Nikolaidis and Ioannis Pitas
Department of Informatics
Aristotle University of Thessaloniki, GREECE
Email: {tefas,nikolaid,pitas}@aiia.csd.auth.gr

***Abstract*—In this work we proposed the use of local binary patterns in combination with double spectral analysis for facial image clustering applied to 3D (stereoscopic) videos. Double spectral clustering involves the fusion of two well known algorithms: Normalized cuts and spectral clustering in order to improve the clustering performance. The use of local binary patterns upon selected fiducial points on the facial images proved to be a good choice for describing images. The framework is applied on 3D videos and makes use of the additional information deriving from the existence of two channels, left and right for further improving the clustering results.**

## 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 non-overlapping subsets (clusters) $\mathcal{C} = \{C_i | C_i \subseteq \mathcal{P}\}$. 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 feature films, where the goal is to separate facial images derived through face detectors and trackers [1]–[3] into groups, for which within-cluster similarity is high whereas between-cluster similarity is smaller. Ideally, such an algorithm should assign all facial images from a certain subject to the same cluster. A few papers ( [4]–[8]) deal with facial image clustering in single view (2D) videos, and some of them [7], [8] utilizes spectral clustering approaches, but work on stereoscopic videos is extremely limited. In most papers dealing with single view videos, the facial image dataset is derived from short videos like in [5]. The method proposed in [8] is tested on a feature film which consists of a single shot where characters appear in consistent clothing. The stereoscopic facial image clustering method presented in [9] is also applied in short videos. In contrast, the proposed method is applied on 3 full length feature films.It should be noted that a number of facial image clustering techniques [10], [11] are applied on still facial images, i.e. facial images ont derived from films, but in this case the requirements are different. Usually this means a more confined environment, less illumination, pose and expression variation and lack of non-facial images.

The rest of this work is organized as follows: Section 2 contains a short description of the local features that have been used to describe facial images, namely the Local Binary Patterns ($LBP$s) and their variations as well as the way fiducial points are calculated. Section 3 provides the problem statement and a summary on spectral clustering while Section 4 introduces the double spectral clustering and the improvements proposed. Section 5 contains the experimental evaluation procedure and results Section 6 concludes the paper.

## II. Local features for the description of images

### A. Local Binary Patterns

In this work, Local Binary Patterns ($LBP$) [12] have been used as image features. $LBP$s have been used in various works due to their good properties such as robustness in illumination changes, simple and fast calculation and effectiveness in performance. Furthermore, as they are local descriptors they suffer less from partial occlusion [13]. The $LBP$ for a certain pixel $(x_c, y_c)$ with grayscale intensity $g_c$ is given by the differences of the central pixel with the $P$ neighboring ones in prespecified order. The $LBP$ of this pixel is then given by the sum of the quantized differences weighted by $w_i = 2^{i-1}, i = 1, .., P$. The formula for $LBP$ calculation is: $LBP_{P,R}(x_c, y_c) = \sum_{i=1}^{P} s(g_i - g_c)2^{i-1}$ where $R$ is the radius of the neighbourhood, $s(z)$ is 1, if $z \geq 0$ and 0, if $z \geq 0$. The standard approach involves the calculation of these $LBP$s for various pixels or even all possible pixels and afterwards calculation of a histogram using these LBP values. Then using these histograms as features, similarities can be calculated.

Besides $LBP$ we have also used $CS\text{-}LBP$ [14] and thresholded $CS\text{-}LBP$ which we denote as $tCS\text{-}LBP$. The $CS\text{-}LBP$ is defined as $CS\text{-}LBP_{P,R}(x_c, y_c) = \sum_{i=1}^{P/2} s(g_i - g_{P/2+i})2^{i-1}$ where $s(z)$ is the same thresholding function as in the standard $LBP$. The $tCS\text{-}LBP$ uses the same formula as $CS\text{-}LBP$ but with a different flexible threshold [15]: $s(z) = 1$ if $z \geq tm$ and 0 if $z < tm$, where $m = \frac{1}{P+1}(g_c + \sum_{i=1}^{P} g_i)$ and $t \in [0.01, 0.1]$. In order to increase the effectiveness of the LBPs as features extractors they should be applied in specific facial fiducial points. To this end, a facial fiducial point detection method has been used as described in the following section.

### B. Facial fiducial points detection

In order to cope for pose and perspective variations that appear when dealing with images derived from feature films we calculated the $LBP$ features on a number of fiducial points instead of calculating them on the entire facial image. Two passes of fiducial point detectors were used. The first one is for the calculation of 66 points, such as outline of eyes, eyebrows, mouth etc, [16] and the second one [17] for better localization of these points. We then perform alignment and scaling of all facial images with respect to certain of these fiducial points. By this way we end up with 66 roughly aligned points. $LBP$ descriptors are then calculated upon patches around these points. Then a histogram with $K$ bins is calculated for each of these features. By this way a descriptor of dimension $66 \times K$ is calculated for each image. The similarity among images is calculated by evaluating the $\chi^2$ distances between each of the 66 histograms in the 2 images and summing them up in order to come up with a single distance value $d_{ij}$. By using the above approach a similarity matrix is produced for the set of facial images and can be used with spectral clustering, as explained in the next Section.

## III. Problem statement and spectral clustering

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:

$$\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} \qquad (1)$$

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 [18].

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) [19] described in subsection III-A while another solution could be the approach introduced in [20] 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 [19].It is proven in [19] 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 approach in [20]

While Ncut solves a binary problem, the spectral clustering approach proposed in [20] solves a multiclass clustering problem. This solution also uses the Laplacian matrix $\mathbf{L}$ defined in (1). 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 [21] to these new samples.

## IV. 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 [19] 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 [20] 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 [20]. 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 (1).

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 [20], 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:

$$[\mathbf{W}']_{i,j} = \frac{1}{(\mathbf{y}_i - \mathbf{y}_j)^T (\mathbf{y}_i - \mathbf{y}_j)} \tag{2}$$

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.

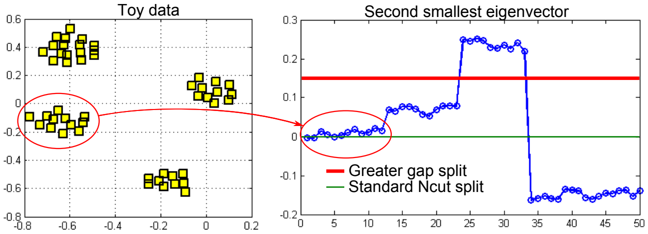
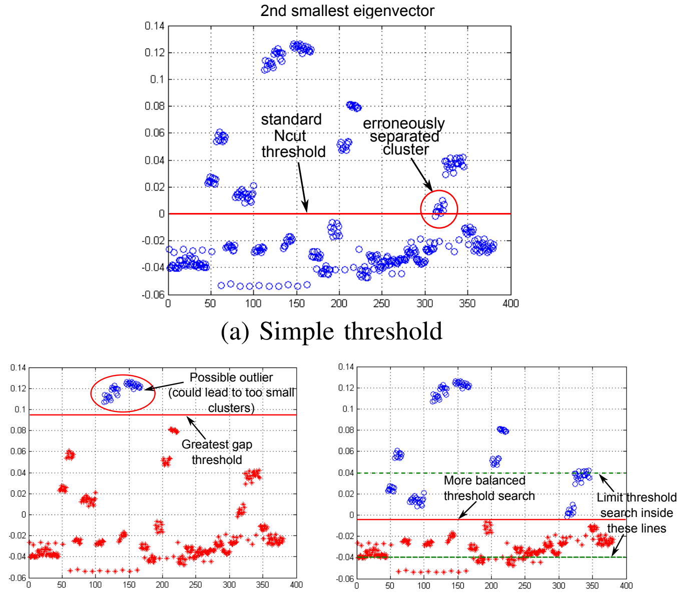


Fig. 1. Ncut unsuccessful clustering in toy example

### 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 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 $v$: $Limits: \pm av,\ a \in (0, 1]$.



(a) Simple threshold

(b) Overall greatest gap threshold

(c) Limited greatest gap threshold

Fig. 2. The effect of using different thresholds

## 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 [1] is performed every $n$ frames in one channel (video) of the stereoscopic video, followed by a face tracker [22] at the same channel. In the second approach, face detection [1] was applied on both channels, mismatches between the two channels were rejected and a stereo tracking algorithm [3] 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. In our approach each of these trajectories is represented by a single facial image.

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.

### B. Experimental results performance

Experiments have been conducted using all three variation of $LBP$s but we present (Table II) only the best performance due to limited space. Of the three variants the $CS$-$LBP$ and $tCS$-$LBP$ showed a more consistent performance but with small performance differences over $LBP$. In the case where

TABLE I. VARIOUS CHARACTERISTICS OF THE THREE FEATURE FILMS

| Size of cast | # of frames | # of trajectories | |
|---|---|---|---|
| | | stereo | single view |
| Feature film #1 | | | |
| 22 | 181,763 | 1246 | 1545 |
| Feature film #2 | | | |
| 47 | 150,361 | 1222 | 1559 |
| Feature film #3 | | | |
| 58 | 196,224 | 1726 | 2238 |

TABLE II. RESULTS ON THE 3 FEATURE FILMS FOR VARIANTS OF NCUT AND SINGLE VIEW & STEREO VIDEOS (GG: GREATEST GAP)

| Feature film #1 | | | |
|---|---|---|---|
| | Left | StereoL | StereoLR |
| Ncut | 0.4583 | 0.5815 | 0.5864 |
| Double spectral | 0.6207 | 0.6305 | 0.6566 |
| GG Double spectral | 0.6443 | 0.6777 | **0.6815** |
| Feature film #2 | | | |
| | Left | StereoL | StereoLR |
| Ncut | 0.3771 | 0.4205 | 0.4415 |
| Double spectral | 0.4725 | 0.5184 | 0.5573 |
| GG Double spectral | 0.4904 | 0.5368 | **0.5673** |
| Feature film #3 | | | |
| | Left | StereoL | StereoLR |
| Ncut | 0.3104 | 0.3257 | 0.3914 |
| Double spectral | 0.4955 | 0.5201 | 0.5673 |
| GG Double spectral | 0.5157 | 0.5451 | **0.5762** |

both stereo channels (left-right) are being used we have an option of 4 different options to calculate similarities between facial trajectories $i$ and $j$, namely $L_i - L_j$, $L_i - R_j$, $R_i - R_j$ and $R_i - L_j$, where $L$ and $R$ denote the L(eft) and R(ight) channel respectively. For example, $L_i - R_j$ means that the similarity is calculated using the left facial image of trajectory $i$ and the right facial image of trajectory $j$. Of these 4 scores, the maximum one was used as similarity score between the trajectories.

Results are presented 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 GG threshold, the latter achieving the best performance The use of stereo information in detector and tracker (Table II, column 'StereoL') also improves the performance and provides an extra way to further improve the results. Also, when both channels are being used for clustering (Table II, column 'StereoLR'), as described in previous paragraph, an additional significant improvement is observed.

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. In order to focus the evaluation on the purity of clusters we used the following approach: we oversplitted the set to a number of clusters larger than the one needed and then merged them in a way a user would merge them, i.e. by merging all clusters in which the majority of the samples belong to the same class.

## VI. CONCLUSION

In this work we proposed a new framework for facial image clustering in 3D videos which includes the following novel elements: a) use of local $LBP$s on fiducial points for image similarity calculation, b) the use of double spectral techniques to further improve an already efficient spectral clustering technique and c) use of additional information derived from stereo videos to further improve performance. Combining all these three elements has proven to achieve very promising results. Future work will be directed towards further exploiting stereo information for improving the clustering results.

## ACKNOWLEDGMENT

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.

## REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] G. Orfanidis, N. Nikolaidis, and I. Pitas, "Facial image clustering in 3d video using constrained ncut.," *EUCIPCO*, 2013.

[10] 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.

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

[12] T. Ahonen, H. Abdenour, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 2037–2041, 2006.

[13] A. Rama, F. Tarres, L. Goldmann, and T. Sikora, "More robust face recognition by considering occlusion information," *Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pp. 1–6, 2008.

[14] M. Heikkil, M. Pietikinen, and C. Schmid, "Description of interest regions with center-symmetric local binary patterns.," *Computer Vision, Graphics and Image Processing*, pp. 58–69, 2006.

[15] X. Wu and J. Sun, "Improved region local binary patterns for image retrieval.," *Advances in Computer Science and Information Engineering*, pp. 283–288, 2012.

[16] A. Asthana et al., "Robust discriminative response map fitting with constrained local models.," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[17] M. Ui, V. Franc, and V. Hlav, "Detector of facial landmarks learned by the structured output svm.," *VISAPP 2012*, pp. 547–556, 2012.

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

[19] 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.

[20] 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.

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

[22] S. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters.," *IEEE Transactions on Image Processing*, vol. 13, pp. 1491–1506, 2004.