Facial image clustering in stereoscopic videos using double spectral analysis

Georgios Orfanidis *, Anastasios Tefas, Nikos Nikolaidis, Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki, Greece

**Abstract**

In this work, we are focusing on facial image clustering techniques applied on stereoscopic videos. We introduce a novel spectral clustering algorithm which combines two well-known algorithms: normalized cuts and spectral clustering. Furthermore, we introduce two approach for evaluating the similarities between facial images, one based on Mutual Information and other based on Local Binary Patterns, combined with facial fiducial points and an image registration procedure. Ways of exploring the extra information available in stereoscopic videos are also introduced. The proposed approaches are successfully tested on three stereoscopic feature films and compared against the state-of-the-art.

**1. Introduction**

Clustering decomposes an existing set of unlabeled objects \( P \) (for example, facial images) into a number of subsets (clusters) \( C_i \) that fulfill the following conditions:

\[
P = \bigcup_{C_i \in C} C_i
\]

\[
\forall C_i, C_j, i \neq j \in C: C_i \cap C_j = \emptyset.
\]

Essentially, clustering assigns each data sample to a cluster that does not overlap with the other ones. Exceptions to this rule are the fuzzy clustering approaches such as fuzzy c-means clustering [1]. Since clustered data items can be assigned cluster labels that may have semantic meaning (e.g., the person name in the case of facial images) clustering is also related to other tasks, like classification and label propagation [2]. Clustering is an unsupervised technique, while classification is supervised and label propagation is a semi-supervised one. This paper deals with facial image clustering, where the goal is to organize facial images into groups, for which within-cluster similarity is high, whereas between-cluster similarity is small. Although facial image clustering has been researched recently, it has received much less attention than facial image recognition [3]. Ideally, each facial image cluster should contain only facial images of a single person and, if possible, all relevant images of that person. This means that if a person has many visual facial appearances in a video, all these appearances should be gathered in one cluster or, at least, the clusters should contain images from just one person. In other words, the derived clusters should be as homogeneous as possible.

Relatively few papers [4–8] deal with facial image clustering in single view (2D) videos. Some of them [7,8] utilize spectral clustering approaches. However, work on stereoscopic facial image clustering is extremely limited [9] and so are methods that can operate on challenging facial image sets, such as those coming from face detection and tracking in movies. In most papers dealing with facial image clustering in
single view videos, the facial image dataset is derived from short videos, as in [5]. The method proposed in [8] is tested on a feature film which consists of a single shot, where actors appear in consistent clothing thus making clustering relatively easier. The stereoscopic facial image clustering method presented in [9] is also applied in short videos. In contrast, the proposed method has been applied on three full length feature films. Also, a number of facial image clustering techniques [10–13] have been applied on still facial images, e.g., on family pictures or personal photo albums. Methods in [12,13] are based on data association or probabilistic frameworks which follow a different problem formulation in order to achieve a comparable goal. However, the images in such cases are usually acquired in a more confined environment and have less illumination, pose and expression variations. For example in [12] the authors use 2 datasets, one being The Gallagher Collection Person Dataset [14] and the other a modified version of Labeled Faces in the Wild [15], which contains images usually well illuminated and well registered. Moreover, in this case, the datasets do not contain non-facial images (noise). Hence, facial image clustering is easier than in the case of movies, where actors may have unusual expressions, different poses or even makeup. Furthermore, in the case of movies, facial image and general actor appearance characteristics (e.g., clothing) may change drastically from scene to scene.

This paper deals with facial image clustering applied on stereo video content coming from 3D movies, where the goal is to organize facial images derived through stereo face detection and tracking into clusters. In order to improve the clustering performance quality we have experimented with a clustering technique which combines elements of spectral clustering and Normalized Cut (which also contains spectral clustering elements). This new method is applied on both stereoscopic and single channel videos, in order to evaluate the stereo clustering performance in comparison to the single channel one, and it is also combined with various improvements and variations on specific aspect of the algorithm. These improvements include a dynamic threshold in Normalized Cuts which is compared to a standard fixed one as well as experiments using various facial image as trajectories representatives. Two different facial image similarity measures are being used, one global based on Mutual Information between images and other local based on Local Binary Patterns evaluated around specific fiducial points and combined with an image registration procedure. Extensive experiments on three 3D full feature films have been conducted to measure the clustering performance.

In this work, we are focusing on facial image clustering techniques applied on stereoscopic videos. Towards this end the paper provides various novel contributions that can be summarized as follows:

1. Use of stereoscopic information through the use of stereo face detection and tracking procedure and using both facial images (right/left channel) when calculating similarities.
2. A novel LBP-based similarity measure including facial fiducial points.
3. Use of a double spectral clustering approach involving normalized cuts and spectral clustering.
4. Use of a flexible (dynamic) threshold for better performance of Ncut.
5. Facial image trajectories represented by
   - A single facial image.
   - Multiple facial images sampled at fixed temporal intervals from the trajectory.

There is not a single main contribution but rather multiple ones. If however the criterion to judge the most important (main) contribution is the improvement in the performance of the algorithm, then results show that the influential contribution is double spectral clustering in comparison with Normalized cut.

The rest of this work is organized as follows. Section 2 describes the overall facial image clustering methodology. Sections 3 and 4 contain a short description of the image similarity measures, namely the Mutual Information (MI) and the Local Binary Patterns (LBP) respectively and their variations, as well as the way fiducial points are calculated. Section 5 introduces the double spectral clustering and the introduced clustering improvements. Section 6 contains the experimental evaluation procedure and results, while Section 7 presents conclusions.

2. Facial image representation and clustering

In image clustering, the first essential step is the selection of an image similarity measure $w_{ij}$ between images $i$ and $j$. The images can be considered as vertices $V$ in a similarity graph $G = (V, E)$, while the image similarities are the edge weights $E$. Thus, for $N$ images (facial images in our case), we can create an $N \times N$ similarity matrix $W$. Many clustering approaches are based on such a similarity matrix but there are also techniques such as $k$-means [16] that do not. As a pairwise facial image similarity must be calculated, the complexity of the overall calculation of matrix $W$ is of order $O(N^2)$. Several similarity measures can be used, some of them being quite computationally demanding [7,9,17]. Most similarity evaluation approaches involve the calculation of a feature vector $x_i$ of reduced dimensionality $L$ for each facial image $i$, $i = 1, \ldots, N$. As $L << N_x \times N_y$, where $N_x$ and $N_y$ are the facial image dimensions (in pixels), usually the calculation of the similarity matrix $W$ with $w_{ij} = 1/d(x_i, x_j)$ using a facial image feature distance measure $d$ (and not the original image data) is relatively fast, though still having $O(N^2)$ complexity.

Two different approaches for calculating the pairwise facial image similarities and construct the similarity matrix $W$ are presented in this paper. In the first proposed approach, we used Mutual Information (MI), whereas in the second approach, we used the Local Binary Patterns (LBPs) as facial image features.

Assuming that $W$ is available and symmetric, i.e. $W = W^T$ or equivalently $w_{ij} = w_{ji}$, we then use spectral clustering techniques [18] to perform facial image clustering. The unnormalized Laplacian matrix $L$ and the normalized Laplacian $L_{sym}$ [18] are typically used in spectral clustering:

$$L = D - W, \quad D = \sum_j W_{jj}$$  \hspace{1cm} (3)

$$L_{sym} = D^{-\frac{1}{2}}L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}W D^{-\frac{1}{2}}$$  \hspace{1cm} (4)
We chose to work with the normalized Laplacian $L_{sym}$, as it was proven to have greater merits for clustering than the unnormalized one [18].

3. Facial image similarity evaluation using Mutual Information

In this section, we present in brief the Mutual Information (MI), which can be used as a global similarity measure between facial images. Let $p(x)$ and $p(x,y)$ be the probability density function and the joint probability density function of random variables $X$ and $Y$, respectively, which in our case represent facial images. The entropy $H(X)$ of a random variable $X$ and the joint entropy $H(X,Y)$ of two random variables $X$, $Y$ are defined as

$$H(X) = -\sum (p(x) \log (p(x)))$$

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

Then, the Mutual information $I_{MI}(X,Y)$ of two distributions $X$, $Y$ is defined as

$$I_{MI}(X,Y) = H(X) + H(Y) - H(X,Y) = -\sum (p(x,y) \log \left( \frac{p(x,y)}{p(x)p(y)} \right)).$$

The Normalized Mutual Information ($INMI$) used in this work is defined as [7,19]

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

NMI $INMI(X,Y)$ is used in this work since it has been proven to be more robust to image translation than MI $I_{MI}(X,Y)$ [19]. NMI $INMI(X,Y)$ is calculated on facial images using the HSV color space, as it has been proven to be more robust in illumination changes, compared to RGB color space. A Hue–Saturation normalized MI was defined in [20]

$$INMI(X,Y) = \frac{H(I_1) + H(S_1) + H(H_2) + H(S_2)}{2H(I_1,S_1,H_2,S_2)}.$$  

where $H_I$ and $S_I$ are the Hue and Saturation of each facial image. An approach similar to [7] was used for the calculation of the 4D joint histogram $p(I_1,S_1,H_2,S_2)$. MI and NMI are global similarity measures that are relatively robust to geometrical image transformations [19]. Thus, no further effort, e.g., for facial fiducial point detection, or for image registration prior to the facial image similarity calculation, is required.

4. Facial image similarity evaluation using Local Binary Patterns

Local Binary Patterns (LBP) [21] have extensively been used for facial image representation [21–23]. Their popularity is the result of their good properties, such as robustness in illumination changes and simple/fast calculation. Furthermore, as they are local descriptors they suffer less from partial occlusion [24]. The LBP for a certain pixel $(x_c,y_c)$ with greyscale intensity $I(x_c,y_c)$ is given by the differences of the central pixel with the $P$ neighboring ones, calculated in prespecified order. The quantized differences are weighted by $w_i = 2^{i-1}, i = 1, \ldots, P$, resulting in

$$LBP_{P,R}(x_c,y_c) = \sum_{i=1}^{P} s(I(x_i,y_i) - I(x_c,y_c))2^{i-1},$$  

where $R$ is the radius of the neighborhood and $s(z)$ is a thresholding function, which serves as a binary quantizer for the difference between the central pixel $(x_c,y_c)$ and its surrounding ones $(x_i,y_i)$:

$$s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0. \end{cases}$$

The standard approach involves the calculation of LBPs on various or even all image pixel locations. Furthermore, a histogram using these LBP values is calculated, which

![Fig. 1. LBP calculation in facial images: (a) calculation of LBPs in the entire facial image and creation of one image histogram; (b) calculation of LBPs in specific facial points and creation of one histogram for each facial fiducial point.](image-url)
characterizes each facial image, as shown in Fig. 1. Then, the
LBP histograms can be used as features vectors $x_i$ to calculate
facial image similarities $w_{ij} = 1/d(x_i, x_j)$, where $d$ is an appro-
priate histogram chosen distance metric, e.g., the Euclidean
distance $||x_i - x_j||_2$.

Two different LBP variants, namely CS-LBP [25] and
thresholded CS-LBP (tCS-LBP) have been also considered.
tCS-LBP uses the same formula as CS-LBP but combined with
a different flexible threshold $v(z)$ [23]. CS-LBP is defined as

$$
\text{CS-LBP}_{p,R}(x, y) = \sum_{i=1}^{P/2} s(g_i - g_{P/2 + i})2^{i-1}
$$

We define tCS-LBP as

$$
\text{tCS-LBP}_{p,R}(x, y) = \sum_{i=1}^{P/2} t(g_i - g_{P/2 + i})2^{i-1}
$$

and

$$
t(z) = \begin{cases} 
1, & z \geq m \\
0, & z < m
\end{cases}
$$

where $m = \frac{1}{p+1} \sum_{i=1}^{p} g_i$ and $t \in [0.01, 0.1]$.

CS-LBP produces much smaller binary codes for the same size
neighborhood $R$. For example for a neighborhood with para-
eters $R=1$ and $P=8$, as shown in Fig. 1(b), the binary codes
produced are in the range $[0, 15]$, whereas standard LBP
produces binary codes in the range $[0–255]$. We have also
tested tD-LBP [23], but the experimental results were mediocre.

4.1. Facial fiducial point detection

In order to increase the efficiency of the LBPs as image
representation features, they should be applied on specific
facial fiducial points, instead of the entire facial image. The two
methods for the LBPs histogram calculation are demonstrated
in Fig. 1. The latter approach should be able to deal better with
the variation of facial poses and expressions. To this end, a
facial fiducial point detection method has been used. In the first
step, all facial images are resized to a common average size of
$N_M \times M_M$. Then, a fiducial point detector is used in each facial
image. The Robust Discriminant Response Map Fitting (DRMF)
with constrained local models [26] has been used to this end. It
returns 66 fiducial points placed on the outline of the detected
face (17 points), the outline of both eyes (6 points for each eye),
both eyebrows (5 points each one), the nose vertical axis and
horizontal outline (4 and 5 points respectively) and finally the
mouth outline. For better fiducial point detection and in order
to take care of misregistered facial images, the initial detection
region is enlarged by 50% along each $x,y$ direction. By using an
enlarged ROI, we aim at two goals: (a) better fiducial point
detection and (b) enough space for LBP patch calculation. If the
enlarged ROI exceed the frame bounders, it is padded with the
values of the last row or column. The LBPs, which are
calculated using a padded patch, may not be equally informa-
tive as the ones calculated on a non-padded one. However,
such a calculation is preferable in cases where the LBPs cannot
be calculated, due to out-of-image patch coordinates.

![Fig. 2. Comparison of LBP and CS-LBP features. (a) Calculation of LBP features; (b) calculation of CS-LBP features; (c) LBP histogram calculation.](image-url)
The detected fiducial points define a new Region Of Interest (ROI), in which a second fiducial point detection method is applied, using the Flandmark approach proposed in [27]. This detector returns 7 points plus 1 (eyes and mouth corners, nose and facial center). The facial fiducial points returned by the second method have been proven to depend more on ROI parameters, i.e., its width, height, position. We take advantage of this fact, in order to achieve better facial fiducial point localization by providing slightly translated facial ROIs to the second fiducial point detector. By doing so, we get a different point estimation for each slightly translated ROI. The accumulation of all these estimated points gives a distribution of fiducial points. The final fiducial points are chosen by using the marginal median of all these detected positions [28], which suppresses facial fiducial point outliers.

Below we summarize all the steps involved in the LBP feature calculation:

1. A facial image ROI is returned by the facial image detector/tracker.
2. ROI is being enlarged so as to double its dimensions (quadruple the area it covers). This is done for better fiducial point calculation and to take care of misregistered images. Obviously, the new enlarged ROI contains the actual face plus elements from the background around it.
3. The previous step can lead to ROIs exceeding the frame boundaries. In this case, and in order to be able to deal with LBP feature calculation we pad the area outside the frame boundaries with the border pixels of the image.
4. (First fiducial point detector) This enlarged ROI is used to crop the facial image and provide it as input for the DRMF facial fiducial point detector, which returns 66 fiducial points.
5. We create a new ROI using the positions of the 66 points which is passed as input to the second fiducial point detector we use, Flandmark.
6. (Second fiducial point detector) Since Flandmark is affected heavily by the input ROI, we provide multiple slightly translated versions of the previous step’s ROI in order to create a set of possible fiducial points. We use those points to estimate a final ROI from the marginal median of all these detected positions, which suppresses facial fiducial point outliers. The use of a second fiducial point detector, Flandmark, was chosen in order to improve the robustness of the detected points.
7. The relative translation of this final ROI with respect to the original ROI is used to relocate all 66 fiducial points detected by the DRMF detector to a new position corresponding to the new ROI.

4.2. Facial image alignment and image similarity calculation

After having robustly detected 66 facial fiducial points, we can perform two-step facial image alignment. In the bibliography, e.g. in [12], it is rather typical to align the facial images to be compared by rotating and resizing the image. In order to align the images we use the fiducial points corresponding to the canthi of both eyes in each facial image. In this respect, we used the 4 canthi of both eyes, in order to perform a linear regression to retrieve the parameters of the line connecting both eyes. The linear regression used to derive the axis connecting both eyes is used in order to cope with any fiducial point outlier (any fiducial point corresponding to a canthus that does not fit well in the line connecting the other three points). We have experimented with other methods like the use of only 2 canthi instead of 4 (either the inner or the outer ones), but the results were less consistent. The angle that this line formed with the horizontal axis was used for the rotation of the facial image and of its corresponding facial fiducial points.

We also employ a second alignment step which involves the potential flip of the facial image. This alignment aims to compensate for the variance of the facial image set with respect to the vertical facial axis. Our intention is for all facial images to look towards the same direction, in our case, the left one (arbitrary choice). In order to determine if a facial image requires flipping, we calculate the angle formed between the line that traverses vertically the nose and the vertical axis. If this counter clock-wise angle is positive then the facial image is flipped.

Our final step involves facial image resizing to a common size, so that the LBP features are comparable to each other. It must be noted that this resizing is complementary, since it only deals with slight differences among facial images. In more detail, this step involves face normalization, by rotating and resizing the facial images, in order to get an aligned set of images. At the same time, the coordinates of the initial 66 fiducial points extracted by the DRMF algorithm are also recalculated for the aligned facial images.

In those aligned images there are 66 fiducial points. Several LBP values are calculated in a region around each fiducial point (patch). Then, an LBP histogram with $N_{b}$ bins can be calculated for each fiducial point. In this way, every image has a descriptor comprised of the 66 $N_{b}$-dimensional histograms. The LBP histogram calculation pipeline on specific fiducial point patches is depicted in Fig. 3. The similarity between two images is calculated using the $\chi^{2}$ distance [29] of each histogram pair. The inverted sum of those chi-square distances is a facial image similarity. Using the above approach, a similarity matrix is produced to be used in spectral clustering. The facial image alignment and similarity calculation is summarized in the following steps:

1. (First alignment step) This step involves the use of positions of the 4 canthi (corners) of both eyes in order to estimate the line connecting the eyes. The angle that this line creates with the horizontal axis is used to rotate the facial image and the corresponding fiducial points, so that the line becomes horizontal. A linear regression is being employed in order to estimate the line, which is used to cope with any fiducial point outlier.
2. (Second alignment step) This step involves a possible flipping of the rotated facial image in order for all images to look to the same direction. The decision for the flipping is based on the angle created by the line connecting the fiducial points of the nose with the vertical axis.
3. Finally, on the position of the fiducial points the LBP features are calculated and the corresponding LBP histograms are created.
4. An $\chi^{2}$-distance is used on these 66 points to get an estimation of the dissimilarity of two facial images.
Fig. 3. Fiducial points and LBP feature calculation pipeline.
5. Double spectral clustering and threshold selection for stereo facial images

Two possible spectral approaches to data clustering are the Ncut and the spectral clustering approach in [30]. Normalized cuts (Ncut) were originally used for image segmentation [31]. The algorithm was defined primarily for bipartition, namely for splitting a similarity graph into two parts, which was sufficient for the image segmentation purposes, e.g., for foreground–background separation. However, for facial image clustering problems, where the number of classes is typically significantly larger than two, certain new requirements arise as will be explained later on. Ncut attempts to find an optimal cut between these clusters, which minimizes the edge weights between these two potential clusters and simultaneously maximizes the edge weights within each cluster. The objective function minimized by Ncut is given by

$$\arg\min_{y} \frac{y^T(D - W)y}{y^TD1} \quad \text{subject to} \quad y^TD1 = 0.$$  \hspace{1cm} (15)

The optimal solution to the Ncut problem in (15) is the eigenvector corresponding to the second smallest eigenvalue of \(\mathbf{L}_{\text{sym}}\). Application of a threshold equal to zero upon the elements of this eigenvector results in the separation of the dataset \(\mathcal{P}\) into two clusters. When more than two clusters are required, i.e., as in facial image clustering, either an iterative Ncut application on the dataset is employed, which will use a new eigenanalysis for each iteration, or more eigenvectors are used. The latter solution is valid, because each eigenvector, excluding the first one, theoretically separates different clusters. For example, in Fig. 4, we plot the elements of the first three (2nd, 3rd, 4th) informative eigenvectors in a toy example. It is obvious from Fig. 4(b)–(d) that each eigenvector separates the dataset in a different manner. By combining all three eigenvectors, a clustering \(\mathcal{C}\) with 4 clusters can be produced. The latter is shown in Fig. 5(d). For each eigenvector, a threshold equal to zero was used to separate the samples in two clusters, corresponding to positive/negative eigenvalue entries, respectively. It is obvious, that, even in this toy example, this threshold fails to separate correctly all four classes. More specifically, as shown in Fig. 5(d), 3 vertices obviously belonging to class 4 are erroneously assigned to cluster 1.

The main drawback of the spectral clustering proposed in [30] is the use of k-means, which has limitations, notably the assumption of hyperspherical clusters, the effects of initialization on the clustering performance and the possible convergence to a local minimum. Since Ncut does not assume the above, we opt to use it instead of k-means. We denote this approach that combines the spectral clustering in [30] with Ncut double spectral clustering.

5.1. Double spectral clustering

The proposed double spectral clustering procedure, which is a variation of the method in [30] can be described as follows:

First we present the spectral clustering [30] as it is summarized in [18]
We consider a similarity matrix $W \in \mathbb{R}^{n \times n}$.

- We compute the normalized Laplacian $L_{\text{sym}}$ from $W$.
- We compute the first $k$ eigenvectors $u_1, \ldots, u_k$ of $L_{\text{sym}}$ and construct a matrix $U \in \mathbb{R}^{n \times k}$ which contains the vectors $u_1, \ldots, u_k$ as columns.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the $i$th row of $U$.
- We use Normalized Cuts in order to cluster the points $y_i$, $i = 1, \ldots, n$ into clusters $C_1, \ldots, C_k$ instead of $k$-means as was used in [30].

It must be noted that we do not normalize the samples $y_i$ as in [30], in order to retain as much discriminative power as possible. Furthermore, we create a new similarity matrix $W'$ considering as initial data the set $Y \in \mathbb{R}^{N \times k}$ and using the inverse of the Euclidean distance $||y_i - y_j||_2$ as similarity score:

$$[W']_{ij} = \frac{1}{(y_i - y_j)^T (y_i - y_j)}$$ (16)

Then, the corresponding Laplacian matrix $L_{\text{sym}}'$ is used to apply the Ncut algorithm, in order to split the dataset into 2 subclusters.

Double spectral clustering can be viewed through two equivalent points of view: (a) either as a spectral clustering with final clustering method recursive Ncuts instead of $k$-means (b) or as applying recursive Ncuts to new data samples derived from the original samples via the use of spectral clustering. From the second point of view the general idea behind double spectral clustering is to create a

![Fig. 5. Use of more eigenvector than one to solve multi-class clustering problem: (a) toy example after applying 2nd eigenvector cut; (b) toy example after applying 3rd eigenvector cut; (c) toy example after applying 4th eigenvector cut; (d) the final toy example clustering after applying all 3 eigenvectors cut.](image)

![Fig. 6. Recursive application of Ncut on facial image dataset.](image)
new sample representation that benefits from the eigen-analysis of the Laplacian matrix $L$. This eigen-analysis tends to gather samples which have high similarity scores in the corresponding $W$ and to move away samples having low similarity scores in $W$. The new sample representations have dimension $k$, which equals the number of eigenvectors being used. The choice of $k$ is problem-specific. One rule of thumb is to use the eigenvalue gap, mentioned in [18] and shown in Fig. 8. In this, we can observe that there is an obvious gap in the eigenvector values, after the fourth eigenvalue. Therefore, by using this rule, we end up with retaining 3 eigenvectors. It must also be noted that we do not normalize the samples $y_i$, as in [30] for maximizing discrimination power. Normalization using the $L_2$ norm would mean mapping all samples to a unit hypersphere, which obviously reduces the distance between samples. The final step is to use $k$-means [16] to these new sample representations.

There are two approaches for the creation of more than 2 clusters using Ncut: (a) employing more eigenvectors than just the second one or (b) recursive application of Ncut. We chose the second approach, in order to better control the bipartition of each set and to also achieve better accuracy. After the first application of Ncut, we apply the next Ncut only to a subset of the original set $Y$, as shown in Fig. 6. The criterion for choosing in which subset to apply the Ncut algorithm is defined by the mean inter-subset similarity, which is calculated using the similarity matrix. We have experimented with other criteria, instead of the average inter-subset similarity, like the median inter-subset similarity or trimmed mean or median [28], which exclude extreme data values. However, the mean one gave the best and most robust results. We continue to apply Ncut, until the desired number of clusters has been reached.

5.2. Greatest Gap threshold

As has been previously mentioned and illustrated in Fig. 7(b), the simple choice of zero threshold (in any spectral clustering approach) often fails when used with multi-class problems. For this reason, we propose a new flexible method for threshold selection, which has proven to be more reliable and incorporate this approach in the Ncut algorithm utilized in the proposed variant of [30] (double spectral clustering). The values of the 2nd eigenvector of the toy example illustrated in Fig. 4 are plotted in Fig. 9(a) in ascending order. In this figure, there are four easily separable clusters. However, a fixed threshold usage of zero fails to correctly separate the clusters. More specifically the second cluster is erroneously split into two sub-clusters. The problem is apparent with clusters which get values around zero. These clusters have elements with values close to each other. However, their value may fluctuate around zero axis. The latter fact combined with the hard fixed (zero) threshold leads to wrong clustering, as shown in Figs. 4 and 5.

Instead, we propose to first sort the second eigenvector values and then search for the greatest gap between two consecutive values of this vector, as shown in Fig. 9. Then, we choose as threshold the mean value of these two consecutive values defining the greatest gap (Fig. 9). Fig. 10 demonstrates the recursive application of Ncut to the same toy example used in Fig. 4 with the use of Greatest Gap. Each of the subfigures (a)–(c) in Fig. 5 corresponds to an eigenvector cut using Greatest Gap. The result after three eigenvectors’ cuts is the expected one, that is all classes have separated correctly, in contrast with the result in Fig. 5. Thus, we get a different threshold in each application of the Ncut algorithm, which adapts to each specific eigenvector being used for clustering.
One problem with this approach is that it tends to split the outliers first, as shown in Fig. 11(b), which may not be the optimal clustering strategy. Our experiments have shown that adding some constraints on the Greatest Gap search can be more beneficial. This approach tries to combine two opposite behaviors:

- The zero threshold approach produces quite predictable bipartition in each cluster. The 2 subclusters that are created tend to be of comparable sizes. This is often unacceptable in cases where those subclusters should have been highly unbalanced, e.g. in cases where one subcluster should have 80% of the cluster members.
In our experiments which deal with feature films there are often outliers in the dataset. One such case is shown in Fig. 11 where a subcluster having large values is located in the upper part of the image. Those outliers correspond to extreme, but quite frequent, facial images which for some reason have corresponding features located in larger distance from the others. The search for Greatest Gap in the whole range of eigenvector values tends to first separate these extreme values from the rest. This has proven to be undesirable, at least in an early clustering phase.

To deal with these issues we restrict the range of values where the search for the greatest gap takes place. In more detail, research for the greatest gap threshold within two limits defined as a percentage \( a \% \) of the greatest value occurring in each eigenvector: \( \pm av, a \in (0, 1) \). If \( a = 1 \) the unconstrained (overall) Greatest Gap threshold results, i.e., the threshold search takes into consideration all eigenvector values. This approach has the advantage of providing a robust flexible threshold, without being sensitive to outliers. The two Greatest Gap thresholds corresponding to the same eigenvector in Fig. 11(b)–(c) reveal this fact.

6. Stereo facial image clustering

The stereo information is taken into account by exploiting the left and right channel facial images in the clustering procedure. To this end, we utilize both stereo channels to calculate a stereo facial image similarity, in order to create the similarity matrix \( W \). To do so, we calculate the MI similarity or the LBP-based similarity between two facial images in both left/right stereo channels and we select the maximal one as stereo similarity. That is, four facial image similarities are calculated for each stereo image pair and the greatest one is retained, as demonstrated in the upper right part of Fig. 12.

A similar procedure can be followed when we use multiple images per facial image trajectory, e.g., when we automatically detect and track an actor face in a stereo video sequence, as shown in Fig. 13. In this case a series of facial images over consecutive frames generated through the application of face detection and tracking on one channel is called a facial image trajectory. Unless there are tracking errors, all facial images within such a trajectory belong to the same person. Thus, one image can be used to represent the entire trajectory. Alternatively, each facial image trajectory can be represented by a few images.
of clustering is the number of representatives facial images per trajectory. Experiments have shown that multiple facial image representatives can aid clustering performance. However, using all facial images being detected or tracked could lead to a very large trajectory similarity calculation effort. In this aspect, we choose to use all detected facial images (since these are usually of better quality, i.e. they usually depict frontal facial images well framed within the ROI) and also the facial images generated from tracking lying at the midpoint (in time) between two consecutive face detections, all belonging to the same facial image trajectory. This idea is illustrated in Fig. 13, where a trajectory with 3 face detections and 24 frames in total is demonstrated. Face detection is performed every $n = 10$ frames, followed by a forward/backward face tracking. After the third detection, the ROI is being tracked for only 3 frames, because of an existing short cut. It must be noted that when a single facial image is used to represent a trajectory (see experiments) this image is resulting from a face detection, due to better quality of such images (see above). In order to estimate the similarity between two
stereo facial image trajectories that can have a different number of representative stereo images, we calculate the stereo similarities between all possible combinations of representative stereo facial images and select the maximum similarity as the representative similarity between the two stereo actor/persons appearances. This similarity definition between two stereo facial image trajectories, \( i, j \) each having \( N_i, N_j \) stereo representative images requires the calculation of \( 2N_i \times N_j \) MI or LBP-based image similarities.

In the following sections, we present experiments using both previously described approaches for image similarity calculation, the MI-based similarity measure and the LBP-based one and we also consider two options for the number of facial image representatives (a) just one facial image representative (the first detected facial image) and (b) multiple facial image representatives per trajectory, using the previously mentioned approach for choosing the representative facial images.

7. Experimental evaluation

7.1. Mono and stereo dataset creation

We applied our proposed facial image clustering method in three full length 3D feature films of different duration, number of actors/cast and genre, called Feature film 1, Feature film 2 and Feature film 3 in the respective tables. Face detection and tracking was applied for the creation of the facial image dataset. Although this paper focuses on face clustering and not face detection or tracking, several different approaches were tried for the creation of the dataset, leading to very interesting observations. Indeed, two different approaches were tested, in an effort to highlight the advantage of using stereo data. In the first approach, the face detection \[32]\] is performed every \( n \) frames in one channel of the stereoscopic video, followed by a single channel face tracker \[33]\) at the same channel. This approach corresponds to face detection and tracking applied on a single channel video. In the second approach, face detection \[32]\] was applied on both stereo channels, mismatches between the two channels were rejected and a stereo tracking algorithm \[34,35]\) was applied in both stereo channels. The face detector being used in both approaches, \[32]\), is a frontal face detector meaning that it primarily detects frontal facial images in the input video-frames. Some profile or almost profile facial images are also being detected but their number is quite small compared to the frontal ones. By using the above approaches we end up with a number of facial image trajectories, namely series of consecutive facial image regions. In our approach, each of these trajectories can be represented one or more of representative facial images.

In more detail, in the case of LBP-based clustering the number of image trajectories being processed is slightly smaller than the number of image trajectories derived by the detector/tracker due to the DRMF step. This latter step detects the facial fiducial points on which the LBP features will be calculated. In a few images this fiducial point detection cannot be completed and thus LBP features cannot be extracted. These images are rejected and not included in final dataset. The reason for some images to fail DRMF step is primarily some kind of noise in the image. This includes non-facial images (detector/tracker error), heavily blurred or malformed images etc. Their number though is quite small basically for an analysis on the statistics about rejected and accepted image ratios. Section 7.1 provides some additional information.

It must be noted that the two chosen face detection/tracking approaches create different facial image datasets, with respect to the number of facial images and of facial image trajectories being detected/tracked. Table 1 provides various statistics of the three feature films. The number of actors and frames as well as the number of facial image trajectories per film is provided. For the latter, two numbers are provided, the first one is the number of facial image trajectories which are returned by the detector/tracker and the second number, which is smaller than the first, represents the number of facial image trajectories after being processed by the DRMF (see Section 4 for details about DRMF). Essentially DRMF step acts as a filter to reject non-facial images. Also the number is different for each tracking method used, that is the stereoscopic and single channel tracking, with the facial trajectories generated by the stereo face detector and tracker being fewer in number and longer (in number of frames). This has the advantages of having fewer facial images to be clustered and of producing better actor appearance representations. Yet another issue that can be observed in this table is the number of trajectories that are actually facial image trajectories. This varies greatly among the two methods with the least false positives belonging to the stereo tracking approach. The percentages represent the accuracy of facial image trajectories, that is the ratio of true positives to the whole number of trajectories. Another interesting fact is that DMRF improves the accuracy in both methods, but the improvement is greater in the single channel tracking. The average improvement for the stereoscopic tracking is 5.32% while in the case of the single channel tracking the improvement is more than double, i.e. 11.40%. Another observation that can be made is that the number of actors and, as a result, of the facial image clusters is quite high. This fact increases the clustering problem difficulty. Furthermore, the

<table>
<thead>
<tr>
<th>Feature film #1</th>
<th># of actors</th>
<th># of frames</th>
<th>Stereoscopic tracking</th>
<th>Single channel tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature film #2</td>
<td>1222</td>
<td>1100–90.02%</td>
<td>1559</td>
<td>1235–79.22%</td>
</tr>
<tr>
<td>Feature film #3</td>
<td>150,361</td>
<td>1088–94.28%</td>
<td>1385</td>
<td>1223–88.30%</td>
</tr>
</tbody>
</table>

Table 1
Facial image statistics per trajectory of the three feature films.
cardinality of the facial image clusters varies significantly for the various actors, as some actors (typically the leading ones) appear in many video frames and, hence, are depicted in many facial images, while others have very few appearances. This imbalance imposes additional difficulties in clustering that must be dealt with.

Table 2 also presents per frame statistics for the first feature film. In this table the most interesting part is the comparison between stereoscopic and mono video channels. The initial potential facial images returned by the detector are quite distinct: 78,071 for stereoscopic and 91,844 for mono video which means a difference of 13,773 facial images. But if we analyse more the nature of these potential facial images we get different figures: 72,848 images with actual facial content in stereo in comparison with 79,263 actual facial frames in mono, a difference of 6415 images, less than half of the previous number. The most interesting part is that after the DRMF step we end up with even more converging figures: 72,739 actual facial frames in stereo in comparison with 74,548 facial images in mono, which gives a ratio of 0.9757, almost all facial images detected by the mono tracker approach are included in the stereo tracker approach. Finally, for statistical completeness we have manually annotated all facial occurrences in feature film 1 and came up with a number of 136,230 facial images, which gives an actual true face detection of 53.3% and 54.72% respectively for stereo and mono detection and tracking after DRMF step. The reason for this relatively small number of facial images that are being detected by the face detection and tracking is the use of a frontal face detector which missed most non-frontal faces. It must be noted that the two previous numbers do not correspond to exactly the same frames and are shown here merely as statistic figures.

7.2. Facial image clustering performance measurement

In facial image clustering, we can very easily create a clustering ground truth, in most movies. Therefore, the \( F \)-measure \([36]\) can be used to evaluate clustering performance, by taking into consideration both precision \( p \) and recall \( r \). Given a set \( P \), a certain clustering \( C \) and the clustering ground truth \( C^* \), the recall \( r(i,j) \) of cluster \( C_i \) with respect to cluster \( C_i^* \) is defined as

\[
r(i,j) = \frac{|C_i \cap C_i^*|}{|C_i^*|},
\]

where \( |C_i| \) is the cardinality of the set \( C_i \). The precision \( p(i,j) \) of cluster \( C_j \) with respect to cluster \( C_j^* \), is defined as

\[
p(i,j) = \frac{|C_j \cap C_j^*|}{|C_j^*|},
\]

The \( F \)-measure for the two clusters is defined as

\[
F_{ij} = \frac{2p(i,j)r(i,j)}{p(i,j)+r(i,j)}.
\]

The overall \( F \)-measure is given by

\[
F = \sum_{i=1}^{L} \max_{j=1,...,K} (F_{ij})
\]

and takes values in the range \([0,1]\), with 0 being the worst and 1 being the perfect score.

In the current state-of-the-art, facial image clustering contains errors that should be manually corrected by an annotator, so that the overall results become useful, e.g., in a broadcast or postproduction environment. It has been noted that it is faster for an annotator to merge erroneously split clusters, rather than to “clean” an inhomogeneous cluster by pruning outlying entries. Therefore, we decided to oversplit the facial image dataset in more clusters than needed and expect the user (annotator) to perform a manual merging step, after the clustering result is presented to him. Then the clusters are merged according to the class region identity of the dominant clusters members, e.g., of the cluster majority. Finally, the \( F \)-measure is computed in the merged clusters. In all the experiments we have evaluated the performance of the clustering algorithms using the \( F \)-measure on the merged clusters. The number of user merging actions (measured in clicks) is roughly between 5% and 10% of the number of the representative facial images, which is an easily acceptable load for the archivist.

7.3. Facial image clustering results using Mutual Information

In this section, we present the experimental results of the proposed double spectral clustering approach using the MI similarity for building the facial image similarity matrix. In Table 3, we present the face clustering results and compare the following algorithms: Ncut, the proposed double spectral Ncut and the proposed greatest Gap Double spectral Ncut. We applied those three methods in the two facial image datasets, mentioned in Section 7.1, i.e., on a single channel dataset and on a stereoscopic channel dataset. We used either one or multiple facial image representatives per trajectory, as mentioned in Section 6. The best results are indicated in bold letters. They always correspond to the proposed novel methods. It is obvious that all introduced novel clustering features, namely the use of stereoscopic information, the use of multiple facial image representatives per trajectory, double spectral and greatest gap clustering, enhance facial image clustering performance, leading to almost doubling the \( F \)-measure versus the baseline method.
7.4. Facial image clustering results using LBP

In this section, we present the experimental clustering results, when using LBP-based facial image similarity features. The experiments have been conducted using the same facial image datasets from the three full feature films mentioned in Section 7.1. Also, as mentioned in Section 4, three variations of LBPs were used. Of the three variants, CS-LBP and tCS-LBP showed a more consistent clustering performance, but with small performance superiority over LBP. On the other hand, if the number of bins is taken into consideration, there is big difference among the three LBP variants. Since CS-LBP and tCS-LBP differ only in the thresholding function (11) and (14), in the subsequent analysis we shall consider only the LBP and CS-LBP variants. In the case where the neighborhood parameters are \( R = 1 \) and \( P = 8 \), LBP produces values in the range \([0\text{–}255]\), while CS-LBP only produces values in the range \([0\text{–}15]\). Thus, CS-LBP is a good choice since it produces a much smaller feature values’ range, without having a performance deficiency [25]. As regards the number of the histogram bins, we could choose any value up to 256 for LBP and 16 for CS-LBP. Experiments have shown that 16 bins is a good choice for CS-LBP, while 256 is not such a good choice for LBP. Instead, we opt for the use of \( k = 24 \) bins for LBP. In Table 4 we present the effect of the number of bins on the modified F-measure in LBP-based similarity facial image clustering results. All experiments reported in this table are conducted in the stereo left channel only. The results are quite similar for other facial image datasets. In this table it is obvious that the choice of \( N_b = 24 \) bins (odd columns), outperforms in all cases \( N_b = 256 \) bins, even columns. Furthermore, we experimented with various parameters and chose to use a neighborhood of \( P = 8 \) for all cases, a varied radius of \( R = 10, 16, 25 \), and the \( \chi^2 \) histogram distance. Also the number of facial fiducial points was fixed to 66, which is equal to the number of fiducial points derived from the DRMF algorithm.

The use of the second facial fiducial point detector described in Section 4.1 in order to improve the accuracy of the detected points, and thus to better final facial image clustering results since our approach is based heavily on the robust facial fiducial point detection. In order to verify the validity of this approach we have conducted two experiments, one based on the use of a single facial fiducial point detector (DRMF) and the second based on two facial fiducial point detectors (DRMF plus Flandmark). Results are presented in Table 5 and reveal a clear superiority of the two detectors.
approach. In each of the 3 feature films the method using two facial fiducial points detectors had much better facial image clustering results, thus verifying our choice. Both experiments use the same parameters: CSLBP with $R = 16$ applied on stereo left channel.

A series of experiments have been conducted in order to evaluate the use of the second alignment step, i.e. the flipping step. Results from these experiments are shown in Table 6. In this table it is evident that the use of this alignment step further improves the clustering performance.

The experiment parameters are the same for both experiments and also with the ones in Table 5: CSLBP with $R = 16$ applied on stereo left channel using DRMF plus Flandmark.

Overall LBP-based facial image clustering performance (modified $F$-measure) results are presented in Table 7. Results refer to the best performing LBP variant, 24, 16 and 16 bins for LBP, CS-LBP and tCS-LBP respectively, two fiducial point detectors and two alignment steps. It can be observed that there is a significant improvement over Ncut by using either double spectral clustering or double spectral...
clustering with constrained greatest gap threshold, the latter achieving the best performance. Also, when both stereo channels are being used for clustering, as described in Section 6, an additional significant improvement is observed.

7.5. Results discussion and method comparison

In this work, two facial image similarity approaches have been used, namely, the MI-based and the LBP-based ones. Each one of them has its pros and cons. First, the MI-based similarity is a global facial image similarity measure. This has the advantage of a relative robustness against facial image geometrical transformations and/or deformations. Therefore, no facial image registration is absolutely necessary. On the other hand, this same feature of MI-based similarity tends to presents its limits when those deformations exceed a certain degree. Since in the MI-based approach no further process is applied, the performance of this clustering techniques is based heavily on the detector/tracker robustness and to the difficulty of the film to be clustered. However, MI calculation is relatively slow. In Fig. 14 some cluster examples (including some clustering errors) using the MI-based approach are presented. It must be noted, thought, that these clusters do not correspond to facial images derived from any of the actual three feature films used in the experiments due to copyright issues with these films, but they derive from a short 3D film in which the LBP-based approach has been applied. Since this is not a feature film, the variation of facial image expressions is somehow limited when compared with the variations found in a typical 3D feature film.

LBP-based features are local similarity measures. This fact can moderate the effect of facial expressions, poses etc., on facial images clustering, since LBPs are extracted on specific facial fiducial points in the facial images. LBP features are also considered robust to illumination changes and occlusion [24], which further improves clustering robustness. Their disadvantage is that facial image model registration is required to find the facial fiducial image points. If this registration fails, or many fiducial points do not appear, e.g., due to self-occlusions, blurred image, the LBP-based methods cannot operate. The fact that correct fiducial point detection affects significantly the performance of the LBP-based approach and thus final clustering results can be seen in Tables 5 and 6 where some steps have been omitted (for further discussion see Sections 4.1 and 4.2). Appropriate parameter selection of the LBP-based approach can help to improve performance but it is more demanding in order to be tuned.

A summary of the performance of the two methods as derived through the experiments are presented below.

MI-based approach does not perform any compensation for any anomaly in facial image detection/tracking. This means that it depends on the nature of the anomaly how it is being dealt with.

- It can cope quite well with facial expressions and rotation variations.
- On the other hand, occlusions tend to be a bigger problem which cannot be dealt easily.

- Slight misregistration of the facial images can be dealt quite efficient by the MI approach.
- Large translation and misalignment of facial image or totally wrong face detections, e.g. a non-facial image, usually cause this approach to fail. The latter is the reason MI fails when a lot of non-facial images are included in the dataset.
- Finally, illumination variations tend to have a noticeable impact on the MI-based approach and affect the clustering results.

The LBP-based approach operates differently in the previous cases, since among other things, this approach involves several steps of alignment, fiducial point correction etc.:

- Misregistration of the facial images, either small ones or larger, can be dealt quite robustly by the use of multiple fiducial point detectors. In general, the method can provide effective corrections as long as the entire face is contained inside the enlarged ROI.
- Larger translation and misalignment of facial images or totally wrong face detections, e.g. a non-facial image often lead to rejection of the image by the DRMF detector since no fiducial points can be found.
- Facial expressions are dealt by the double fiducial point detector and the robust detection of points.
- Occlusions are a difficult case and its effect on the method depends on the degree of the occlusion. Small ones tend to be dealt quite efficiently while larger ones lead to wrong fiducial point detections and failure.
- Illumination variations can be dealt better than in the MI method, mostly because the LBP is quite robust to this kind of phenomena.

Another aspect that was examined in our experiments was the number of representative images per trajectory. Our intuition was that by using multiple images per trajectory we could improve the trajectory similarity evaluation since more images samples are available per trajectory and thus it is more possible to find two images in the two trajectories under comparison that match. Indeed, the inter-trajectory similarity is defined as the similarity between any two samples from the two trajectories as can be seen in Fig. 12 so this intuition seems natural. Regarding the results, some aspects can be pointed out:

- The improvements gained by this approach were less obvious in the MI-based similarity method than in the LBP-based method, which may occur due to the higher discriminant power of LBP features.
- Furthermore, the overall improvement over the single image inter-trajectory similarity was not as high as expected, as the reviewer pointed out.

One possible explanation is that in the case of a single image representative per trajectory this image is usually an image of good quality since it is an image that came from the detector (not the tracker). These images are usually frontal
facial images, the face being well framed within the ROI. On the other hand, when multiple images are being used some of the images are the result of a tracker which is less robust than the detector. In the latter case, if a high score is achieved when comparing just one image per trajectory (due to the good quality of this image) then the additional images do not

Fig. 14. Example of clusters.
influence the score, since it is already high enough and the highest score is kept. A possible solution to this “issue” could be to use a different measure than the highest score (maximum score) between trajectories, for example the median or the mean score of all inter-trajectory scores. But in this case other issues would arise, like the inclusion of relatively misregistered facial images in the limits of each facial image trajectory, and/or the inclusion of optical outliers inside the trajectory, i.e. blurred or occluded facial images, which would affect these “average” scores. Using just the highest score limits these effects. Inclusion of multiple images might provide significant advantages in cases of facial image trajectories with occlusions or blurred images but this also is of limited use, as the detector is possible to ignore highly occluded or blurred images (as they will not be recognized as facial images) and instead opt for more clear facial images.

As regards the facial image clustering performance per se, the LBP-based method is by far better in all experiments, as can be seen in Tables 3 and 7. This points out the superiority of local (LBP) compared to global (MI) image similarity measure. Finally, it should be noted that both the proposed MI-based and LBP-based facial image clustering methods, applied on stereoscopic videos using double spectral and greatest gap threshold techniques achieve at least a 50% increase in facial image clustering performance, if compared with the baseline clustering performance of applying Ncut algorithm on single-channel videos.

### 7.6. Computational complexity and processing times

Another aspect that was studied was the computational complexity of the two methods used for similarity evaluation. Since MI-based similarity can only be calculated for a facial image pair (see Section 2), the similarity matrix calculation in this case has complexity of $O(N^2)$, where $N$ is the number of the facial images of the dataset. On the other hand, the LBP-based similarity method has a complexity of $O(N)$ for the calculation of the LBP’s and the fiducial points and a complexity of $O(N^2)$ for the similarity matrix per se. However, the calculation of the similarity scores between trajectories using the already calculated LBP features involves a rather trivial number of operations for the calculation of $\chi^2$ distance. In other words, the calculation of LBP and more specifically the facial fiducial points detection is the most demanding part but it is performed on $N$ times (not $N^2$). Therefore, in general, the LBP-based method is less computationally demanding.

Table 8 presents the processing times for films containing 150 and 500 frames. As it can be observed by the data in the table, out of the three steps (DRMF and Flandmark algorithm and LBP similarity calculation, the first two are more computational demanding but their complexity with respect to the number of images is $O(N)$ and thus the computational complexity of these steps increases linearly with $N$. It must also be noted that the times shown in Table 8 refer to using both detectors (DRMF and Flandmark) and alignment steps (rotation and flipping). If one detector or one alignment step is being used then the times are further reduced. In addition, if stereo similarity is being used (Fig. 12) times are higher than the presented in Table 8. On the other hand, the MI-based approach has just a single step, the calculation of the MI similarity, which is less time demanding but has to be performed $N^2$ times. Thus, this method is more demanding for a large number of trajectories which is typical of feature films (see the case of 500 trajectories).

### 8. Conclusions

In this work, we proposed a new framework for facial image clustering in stereoscopic videos, which includes the following novel elements: (a) use of two facial image similarity measures, MI-based and LBP-based ones, (b) use of a double spectral clustering approach to further improve an already efficient spectral clustering technique, (c) use of additional information derived from stereo videos to further improve performance and finally, (d) use of multiple facial images per trajectory. Combining all these four elements has proven to achieve very promising facial image clustering results. Future work will be directed towards further exploiting stereo information for improving the clustering results.

### Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007–2013) under grant agreement number 287674 (3DITVS). 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