Face-and-Clothing Based People Clustering in Video Content

Elie El-Khoury
Université de Toulouse
IRIT Laboratory
118, route de Narbonne,
31062, Toulouse, France
khoury@irit.fr

Christine Senac
Université de Toulouse
IRIT Laboratory
118, route de Narbonne,
31062, Toulouse, France
senac@irit.fr

Philippe Joly
Université de Toulouse
IRIT Laboratory
118, route de Narbonne,
31062, Toulouse, France
joly@irit.fr

ABSTRACT

Content-based people clustering is a crucial step for people indexing within video documents. In this paper, we investigate the use of both face and clothing features. A method of extracting a *keyface* for each video sequence is proposed. An algorithm based on the average of the N-minimum pair distances between local invariant features is used in order to resolve the problem of face matching. An original method for clothing matching is proposed based on 3D histogram of the dominant color. A 3-levels hierarchical bottom-up clustering that combines local invariant features, skin color, 3D histogram and clothing texture is also described. Experiments and results show the efficiency of the proposed clustering system.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*

General Terms

Design, Algorithms, Measurement, Performance

Keywords

video people indexing, face, clothing, hierarchical clustering

1. INTRODUCTION

People indexing in video is an important technique for accessing video data effectively as it enables many applications of such as "intelligent fast-forwards" where the video document is browsed by the video sequences containing for example a particular actor or a political leader from the hundreds of shots available for that document.

Generally, content-based video people indexing must pass through different stages: shot boundary detection, people detection, people tracking within shots, people clustering and people recognition (cf. figure 1). The **shot boundary detection** (SBD) is applied as a pre-processing for almost all content-based video retrieval issues. This step aims to break the video data into homogeneous smaller chunks. A detailed review of the state-of-the-art systems can be found in the report of TRECVid [1] [17].

The **people detection** is generally based on face detection. It aims to determine whether or not there are any faces in the image, and if present, return the image location and scale of each face. A recent survey on face detection approaches is available in [10].

The **people tracking** relies on applying non-rigid object tracking techniques as for faces and clothings [11], in order to help detecting people within the shots in cases where the face detector fails. This provides a set of tracks : a track is defined as a sequence of images along which only one person is appearing.

The **people clustering** in video document consists in grouping all tracks that correspond to the same person. This issue was previously studied in [8] where authors proposed a distance metric that is invariant to affine transformation. It was applied for face clustering in order to give an automatic cast listing in movies. Otherwise, people clustering in video is viewed as either a face recognition problem or a classification problem :

- In [1], a system to recognize all the frontal faces of a character in a film using a small set of queries or face exemplars is described.

- In [7], authors propose a classification method by using both visual descriptors based on face and clothing and textual descriptors based on subtitles and transcripts: this allows assigning an automatic name for every face track. A similar work can be found in [13]. But in these works, the distance measure between a pair of face tracks is not given any special attention as clustering methods do.

In this paper, we propose an original method for people clustering using both face and clothing information by describing new matching techniques and by applying a 3-level cascade hierarchical bottom-up grouping method. In order to make our proposed method as workable and portable as possible, many hypotheses are taken into account:

MIR'10, March 29–31, 2010, Philadelphia, Pennsylvania, USA.
Copyright 2010 ACM 978-1-60558-815-5/10/03 ...$10.00.

[1] http://www-nlpir.nist.gov/projects/trecvid/

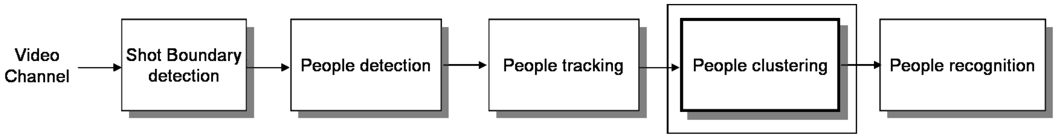

Figure 1: The general architecture of the people indexing system.

- no *a priori* knowledge about the number and the identities of people appearing in the video;
- wide variation in the dimension of the detected face boxes where the lowest resolution may reach 20x20;
- a person may change clothing during the video document. Although it seems very difficult to occur in broadcast news and talk show programs but it is very common in TV series and movies;
- two different people may wear the same costume. This case is very common in team sports video as basketball and soccer;
- lightning conditions may change along the video file.

It is obvious that the face is one of the most reliable descriptors to process the people clustering. But also, other visual features can be helpful like clothing, hair, background information, etc. In the following section, we propose a method for selecting good *keyfaces* used for the matching, and then we present our method for face-based matching which uses SIFT features and we review the skin color matching. In section 3, histogram comparison, dominant color and texture are used for the clothing-based matching. Then, the hierarchical bottom-up clustering that merges all those descriptors is described in section 4. Experiments and results are detailed in section 5.

2. FACE-BASED PEOPLE MATCHING

Face is a very important and discriminant high level feature: the skin color, the geometrical layout, ears, eyes, mouth and nose are descriptors that are often used to recognize people. However, the variations in illumination, partial occlusions, face scale and pose are constraints that make the face-based people matching a difficult task.

In this work, we decide to study the SIFT descriptors as they are known to be highly distinctive and used for object recognition tasks.

Moreover, instead of processing on the whole sequence of faces which is time consuming, we decide to work only on *keyfaces* : for every sequence of frames, we choose one face that must be the most representative one containing the maximum amount of useful information.

2.1 Choice of the keyface

We define a list of criteria that the *keyface* must respect:

1. The area $(w * h)$ of the face box must be as large as possible (cf. figure 2 (a)). In our experiments, we found that the use of $[\min(w,h)]^2$ is slightly better than $w * h$.
2. The ratio of skin part (RSP) within the face box must be as high as possible (cf. figure 2 (b)).

$$RSP = \frac{number\_of\_skin\_pixels}{total\_number\_of\_pixels} \quad (1)$$

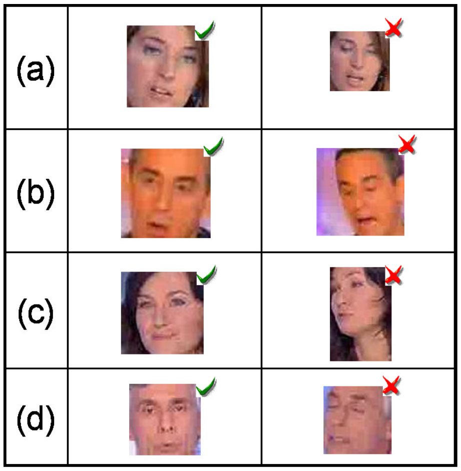

Figure 2: choice of the *keyface*.

3. The width (w) to height (h) ratio must be as close as possible to 3/4 which is used in many face recognition databases [16] [3] (cf. figure 2 (c)).
4. The face must be as frontal and vertically aligned as possible (cf. figure 2 (d)). An indication of the face orientation is given by the image statistical central moment μ_{30} computed on the normalized gray-scale image I_g:

$$\mu_{30} = \sum_y \sum_x (x - x_0)^3 I_g(x,y) \quad (2)$$

where (x, y) are the coordinates of a pixel within the image and x_0 the mean of the abscisses x. The face is frontal and symmetric if μ_{30} is close to 0.

One good way to model the choice of the *keyface* K is to use the following expression :

$$K = \arg\max_k \left(\frac{RSP_k * [\min(w_k, h_k)]^2}{\left(1 + \left|\frac{w_k}{h_k} - \frac{3}{4}\right|\right) * (1 + |\mu_{30}|)} \right) \quad (3)$$

where $k \in [1, ..., N_k]$, N_k is the number of frames within the sequence.

Results detailed in section 5 show that the choice of the keyface using the above formula is widely better than arbitrary selecting the face of the middle frame within the sequence.

2.2 SIFT features

The Scale Invariant Feature Transform was introduced by Lowe in 2004 [12]. SIFT features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in illumination, addition of noise and change in 3D view-point. For more details on the different stages of SIFT features extraction, please refer to [12].

SIFT features are highly distinctive, which allows only few features to be correctly matched with high probability against a large database of features. They are used as baseline features for object recognition in most successful systems like the Columbia university system that gave the best results in TRECVid 2008 evaluation competition [5].

Unlike object recognition systems where huge training on positive and negative data is needed, the face clustering must be done without *a priori* knowledge about the number of people and their identities. Furthermore, the issue for face clustering is not to match between test and template images in order to detect a face in the tested image (as the above object recognition systems do), but the problem is to verify if two faces correspond to a same person or not.

In order to process the matching between SIFT features, we were inspired by the works of [12] and [4].

In [12], the best candidate matching for each keypoint is found by identifying its nearest neighbor in the template image. The nearest neighbor that is defined as the keypoint with the minimum Euclidean distance for the invariant descriptor vector is computed using the Best-Bin-First (BBF) algorithm [2].

Moreover, many features that generally correspond to the background clutter are discarded because they do not have any correct matching in the template image. Lowe proposed an efficient way to get rid of those features by computing the ratio of distances to the closest neighbor and the second closest neighbor in the feature space. This method has some limitations in terms of face recognition. Firstly, the number of selected matchings is not efficient because it depends on the number of extracted keypoints that varies from an image to another: it is more likely to have more matchings between images that both provide great numbers of keypoints than matching between images where at least one of them has a few number of keypoints. Secondly, there is some similarity between faces even though they do not correspond to the same person: there may be matching between features around the eyes, the mouth of different faces. That encourages us to use an additional criteria based on the *minimum pair distance* in order to evaluate this matching.

The *minimum pair distance* was used in [4] to resolve the problem of face recognition and authentication. It consists in computing the distance between all pairs of keypoint descriptors in the test image (I_{test}) and the template image (I_{temp}), and then, it uses the minimum distance as matching score.

$$MPD(I_{test}, I_{temp}) = \min_{i,j}(d(f_i^{I_{test}}, f_j^{I_{temp}})) \tag{4}$$

where the sets of features for test and template images are respectively:

$$\begin{cases} F_{test} = \left\{f_1^{I_{test}}, f_2^{I_{test}}, ..., f_L^{I_{test}}\right\} \\ F_{temp} = \left\{f_1^{I_{temp}}, f_2^{I_{temp}}, ..., f_M^{I_{temp}}\right\} \end{cases}$$

Authors improved their system by 1) matching eyes and mouth and 2) matching on a regular grid. This last matching gave better results since it takes into account the location of the features. The matching between two images is performed by computing the average distance between all pairs of corresponding overlapped sub-images of dimensions 1/4 of width and 1/2 of height. But this method has some weaknesses: since low resolution faces are allowed in our framework, there may be no extracted keypoints in a sub-image. It distorts the average *minimum pair distance*. Furthermore, in some cases, two or more pairs of matched keypoints taken from the same pair of sub-images may be more distinctive than taking only one pair from each.

Our algorithm consists in combining the strong ideas of both [12] and [4] papers.

Firstly, we consider two *keyfaces* $K1$ and $K2$ with the respective set of extracted SIFT features :

$$\begin{cases} F1 = \left\{f_1^{K1}, f_2^{K1}, ..., f_L^{K1}\right\} \\ F2 = \left\{f_1^{K2}, f_2^{K2}, ..., f_M^{K2}\right\} \end{cases}$$

After applying the Lowe's matching in terms of ratio of distances to the first and second closest keypoints in the feature space, a new set of pairs of matched keypoints is provided:

$$P = \{p_1, p_2, ..., p_Q\} \tag{5}$$

where p_i is a pair of features $(f_{i_1}^{K1}, f_{i_2}^{K2})$ and $Q \leq \min(L, M)$.

Secondly, we compute the distance D_{p_i} for each pair of keypoints. Those keypoints are then sorted ascending (i.e. from the minimum to the maximum distance). After that, only the first N pairs are selected to compute their average distance value that we call the "Average of the N-Minimum Pair Distances" ANMPD:

$$D_{sift} = ANMPD = \frac{1}{N}\sum_{i=1}^{N} D_{p_i} \tag{6}$$

This average distance is used as a merging criterion in the hierarchical bottom-up clustering (cf. section 4).

Experiments show that the best value of N is 5 (cf. Table 1). An example of good matching between faces under different conditions using SIFT is shown in figure 3.

2.3 Skin color

Since the face is detected and localized, the goal here is to select, within the face box, the pixels that correspond to the skin part of that face. Two approaches exist in the literature: either modelling the skin color by a trained 2D-Gaussian distribution in the normalized r and b space [18] or using a thresholding method. In our case, the second approach is sufficient because no training is needed and because the processing is done only in the face box, thus we can use wide margin thresholding in order to allow detecting from very light to very dark skin colors.

Therefore, the RGB image is converted to YC_rC_b and HSV systems. Then, thresholding is applied on the C_r, C_b components that are coded on 1 byte, and the hue H is normalized between 0 and 1, using the following expressions:

$$\left.\begin{array}{c} 135 \leq C_r \leq 170 \\ 130 \leq C_b \leq 200 \\ 0.01 \leq H \leq 0.1 \end{array}\right\} \tag{7}$$

Then the matching between the skin colors of two keyfaces is done by computing the distance D_{skin} between their corresponding histograms using the Bhattacharyya expression described in section 3.2.

3. CLOTHING-BASED PEOPLE MATCHING

Since within video documents like debates, TV games, movies and series, a character is wearing the same clothing

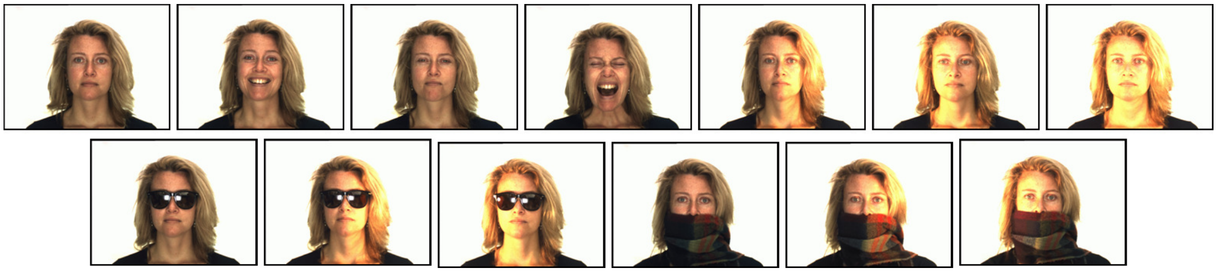

Figure 3: Example of 13 faces of the same person that were correctly matched using ANMPD distance : we can notice different facial expressions, lightning conditions, glasses and occlusions. This example is taken from the AR database [15].

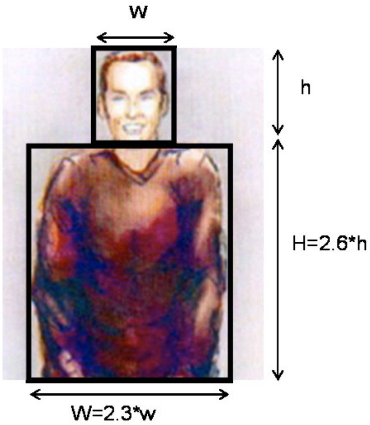

Figure 4: Extraction of clothing using frontal faces.

during all the document or on at least a short period of time (especially for movies), the clustering using clothing information of the person is a crucial solution. In our work, we investigate three clothing descriptors: the 3D histograms, the dominant color and the texture.

3.1 Clothing extraction

Once the face is detected and located, the second goal is to extract the clothing part in order to use it as a matching descriptor in next stages.

For frontal faces, the clothing of the upper-body is extracted as seen in figure 4: the width of the clothing is considered equal 2.3 times the width of the face, and its height equal 2.6 times the height of the face [11].

3.2 Histograms Comparison

The comparison of the 3D histograms of the clothing box is done using the Bhattacharyya distance. This distance is used as a merging criterion in the clustering process. However it can be influenced with some noise due to the background clutter or the foreground occlusions like the examples shown in 5. To eliminate this noise we extract the dominant color and then apply the histograms comparison only on dominant colors.

$$D_{hist}(h_1, h_2) = -\ln\left[\sum_i \sum_j \sum_k h_1(i,j,k) * h_2(i,j,k)\right] \tag{8}$$

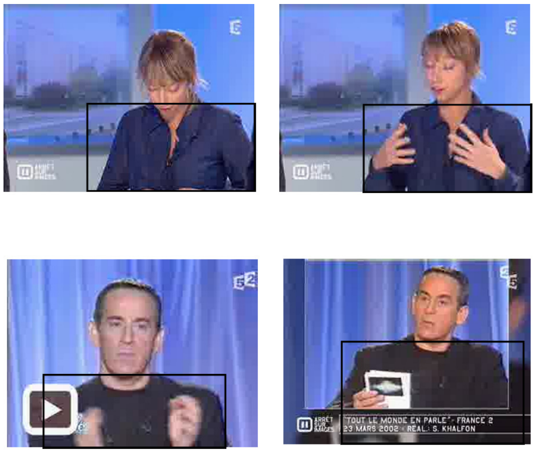
TOUT LE MONDE EN PARLE - FRANCE 2
23 MARS 2002 - REAL: S. KHALFON

Figure 5: Two people with two different costume boxes: the clutters are due to the background and to the foreground objects like hands, characters and logos.

3.3 Dominant Color

The extraction of the dominant color we applied is inspired from the work of [9]. The main difference is that our method considers that the dominant color is spread on a margin of colors in the RGB or HSV space unlike the method used in [9] where the extracted dominant color is a unique triplet of (R,G,B) or (H,S,V) values.

We consider the costume box presented in the image (a) of figure 6.

Five successive steps are done in order to extract the dominant color:

1. In the HSV space, we plot the *Hue* histogram as seen in figure 6.b, then a smoothing process is done in order to eliminate local minima. The maximum value is found on the histogram and its two minimum adjacent neighbors are selected. The most represented hue is located in the margin delimited by those two minima.

2. We return back to the image and we exclude all pixels where the Hue value does not correspond to the selected margin. In figure 6.c , the eliminated pixels are represented in black while the pixels left are illustrated in white.

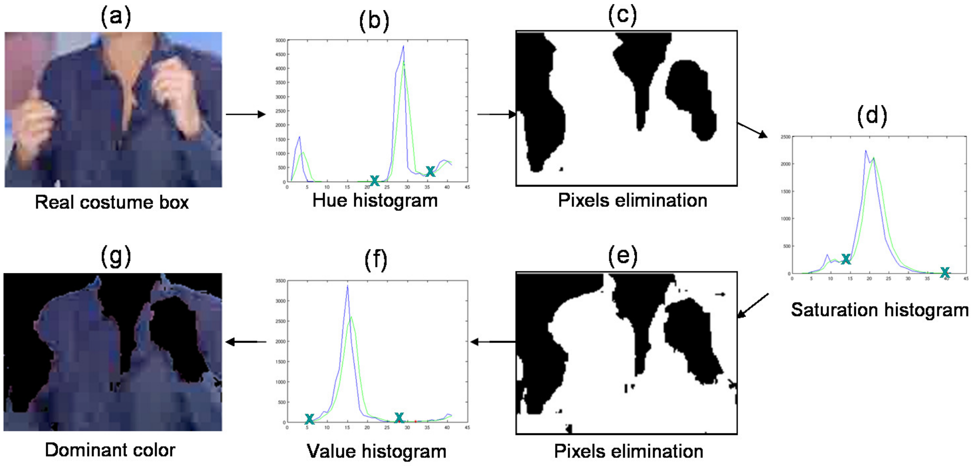

Figure 6: Extraction of the dominant color.

3. On the pixels left, the *Saturation* histogram is computed. Then, the most represented saturation is selected like in step 1. (cf figure 6.d).

4. Again, the pixels that do not correspond to the saturation margin are eliminated as illustrated in figure 6.e.

5. The same process of searching for the most representative value is done (figure 6.f) and the corresponding pixels are selected.

Finally, as seen in figure 6.g, the dominant color is extracted from the image box while the black color corresponds to the eliminated pixels. More examples are shown in figure 7 where the clothing and its dominant color are shown.

3.4 Texture

In this work, we use the Gabor texture feature vector that was introduced in [14]. In order to compute the distance between the textures of two different clothings i and j, we compute the normalized distance in the feature space between the corresponding feature vectors F^i and F^j.

$$\left\{ \begin{array}{l} F^i = \left[f_1^i, f_2^i, ..., f_Q^i\right] \\ F^j = \left[f_1^j, f_2^j, ..., f_Q^j\right] \end{array} \right. \tag{9}$$

The distance is defined by:

$$D_{texture}(i,j) = \sum_q \left| \frac{f_q^i - f_q^j}{\alpha(f_q)} \right| \tag{10}$$

where $\alpha(f_q)$ is the standard deviation of the q^{th} coefficient of the feature vector all over the database.

4. HIERARCHICAL BOTTOM-UP CLUSTERING

After listing the different kinds of face and costume features that can be used to help clustering tracks that correspond to the same person, the issue here is to find an efficient way to combine all those information in order to perform the most accurate clustering. It is obvious that tracks that verify all the merging criteria listed above are favoured to be merged. But in some cases where illumination, background

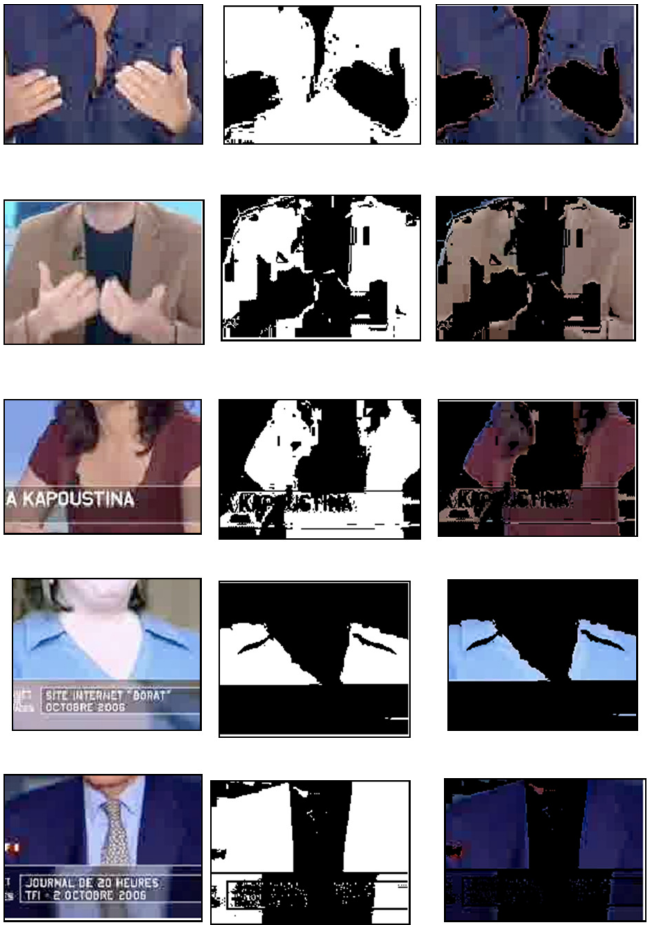

Figure 7: Examples of extracted dominant color areas.

clutter and clothing may change, some of the above criteria will not be verified. In this case, we give more confidence to some special descriptors. That is why we decide to do a 3-levels hierarchical clustering:

- **First-level hierarchical clustering.** This step is illustrated in figure 8. After extracting face and clothing features, distance matrices D_1 (SIFT), D_2 (Skin), D_3 (Histogram) and D_4 (Texture) are reconstructed by computing the appropriate distance between every pair of tracks in terms of the corresponding feature. Then we define a similarity matrix that combines all the above matrices. Every element of that matrix is computed using the following expression:

$$S(i,j) = \prod_{a=1}^{A} \max(Thr_a - D_a(i,j), 0) \qquad (11)$$

where $S(i,j)$ denotes the similarity between the i^{th} track T_i and the j^{th} track T_j where i and j varies from 1 to $N1$ which is the number of tracks. $S(i,j)$ may be even positive if there is good matching or equal to 0 if at least one of the descriptor disagrees the matching. $D_a(i,j)$ is the distance between T_i and T_j in terms of the a^{th} descriptor. Thr_a is the threshold that corresponds to the a^{th} descriptor. It is tuned by processing the clustering method using only this descriptor (cf. table 3). In this study, $A = 4$ since there are only 4 descriptors.

Then, the clustering is done between tracks/clusters that are similar in terms of the resulting similarity matrix. It is done in a hierarchical bottom-up manner, i.e. starting from the most similar tracks/clusters, using the complete linkage property. After each merging between two tracks T_i and T_j, the matrices are updated by eliminating the i^{th} and j^{th} rows and the i^{th} and j^{th} columns and by inserting only a row and a column at the I^{th} position where $I = min(i,j)$ and their elements are computed as follows:

At the position k of the I^{th} row (or column),

- the distance based on the SIFT features of the face uses the single linkage:

$$D_{sift}(I, k \notin \{i,j\}) = min(D_{sift}(i,k), D_{sift}(j,k)) \qquad (12)$$

- the distance based on the skin color of the face uses the average linkage:

$$D_{skin}(I, k \notin \{i,j\}) = \frac{n_i D_{skin}(i,k) + n_j D_{skin}(j,k)}{n_i + n_j} \qquad (13)$$

where n_i and n_j are the number of skin pixels of the i^{th} and j^{th} tracks/clusters.

- the distance based on the color histogram of the clothing uses the full linkage:

$$D_{hist}(I, k \notin \{i,j\}) = D_{bhattacharyya}(H_I, H_k) \qquad (14)$$

where

$$H_I = \frac{n_i H_i + n_j H_j}{n_i + n_j}$$

- the distance based on the texture of the clothing uses the average linkage:

$$D_{texture}(I, k \notin \{i,j\}) = \frac{D_{texture}(i,k) + D_{texture}(j,k)}{2} \qquad (15)$$

The appropriate linkage type for each descriptor is chosen according to the nature and the behaviour of this descriptor.

Consequently, the updated similarity matrix is computed using equation (11). The clustering is repeated until the stopping criterion is verified i.e. when all similarities are equal to 0.

At the end of the clustering, a new set of clusters (N_2 clusters with $N_2 < N_1$) is obtained with their corresponding distance matrices as seen in figure 8.

- **Second-level hierarchical clustering.** After a first clustering where the merging confidence is very high, a second clustering is done in terms of the **clothing similarity**. In this case, two sufficient conditions should be verified:

  - at least one among the two clothing descriptors is working: the second descriptor may fail if there are partial occlusions (the texture descriptor fails!) or lightning variations (the color histogram comparison fails!);
  - at least one among the two face descriptors is working: it is taken into account in order to prevent merging between two people that are wearing the same clothing.

The above constraints are expressed by the following formula:

$$S(i,j) = \max(S_{13}(i,j), S_{14}(i,j), S_{23}(i,j), S_{24}(i,j)) \qquad (16)$$

where

$$S_{ab}(i,j) = \min(D_a(i,j) - Thr_a, 0). \min(D_b(i,j) - Thr_b, 0) \qquad (17)$$

S_{13} is the similarity based on the SIFT features of the face and the histogram of the clothing, S_{14} is the similarity based on the SIFT features of the face and the texture of the clothing, S_{23} is the similarity based on the skin color of the face and the histogram of the clothing, and S_{24} is the similarity based on the skin color of the face and the texture of the clothing.

After each merging between two clusters, the matrices are updated as above. The clustering is repeated until the stopping criterion is reached, i.e. all similarities are equal to 0.

- **Third-level hierarchical clustering.** When the illumination varies or the clothing of the person changes, color-based features and texture features are subject to change. In this case, the only confident features that will remain useful are the SIFT features on faces. That is why a final clustering step must be done according only to SIFT features. This clustering is repeated until the stopping criterion is verified i.e. all similarities are higher than Thr_1.

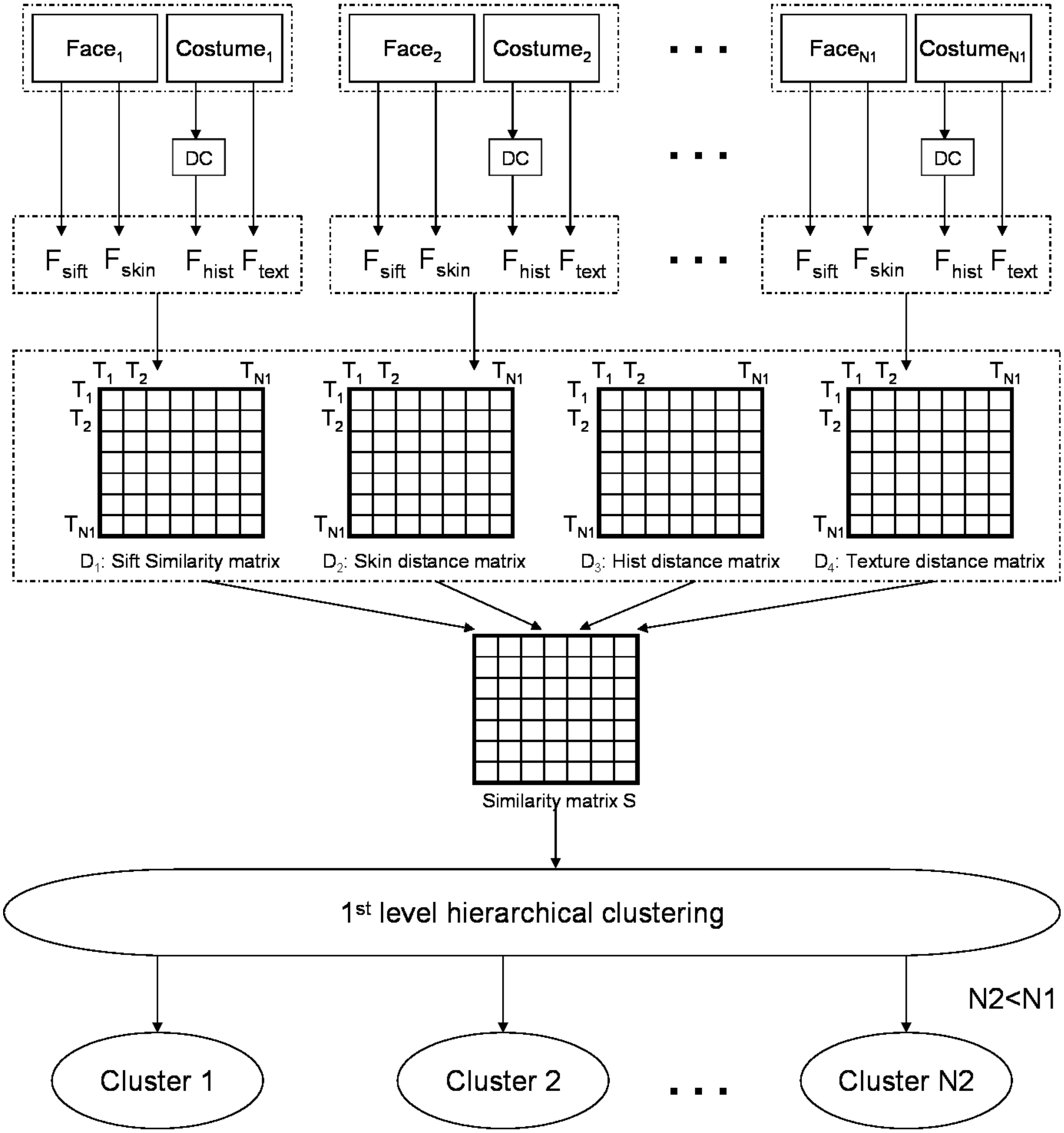

Figure 8: First-level hierarchical clustering.

5. EXPERIMENTS AND RESULTS

5.1 Evaluation tool

In order to mesure the performance of the proposed clustering method, we were inspired from the work done in the speech processing community to evaluate speaker diarization systems. The tool we used is defined by the speech group of NIST[2].

Thus, the performance of the people clustering task is measured according to the errors that occur when person turns of the automatic system do not match the expected person turn in the ground-truth. It means that the error is measured by computing the overall person time that is attributed to the wrong person.

$$E = \frac{\sum\limits_{Allseqs} (dur(seq) * (min(N_R(seq), N_S(seq)) - N_C(seq)))}{\sum\limits_{Allseqs} (dur(seq).N_R(seq))} \tag{18}$$

where for each sequence *seq*:

- dur(*seq*)=the duration of *seq*,
- $N_R(seq)$= the number of people appearing in *seq* according to the **reference** (or ground truth),
- $N_S(seq)$= the number of people appearing in *seq* according to the **system**,
- $N_C(seq)$= the number of **correct** matching, i.e. the number of people appearing in *seq* for whom their matching (mapped) system people are also appearing in *seq*.

5.2 Corpus

The corpus used contains 520 tracks of a *talk show* program of about 40 minutes length where many reports and movie scenes occur. The annotation time for that document took about 12 hours. This is due to the fact that more than one person may appear in the same shot. The total number of the people appearing in this video is equal to 25 : 4 of them appear with two different clothings and 3 others have the same clothing appearance. The resolution of the images is 320x240.

5.3 Results

Six experiments are done in order to test the efficiency of the proposed algorithms.

The first experiment is done in order to choose the best value of N for the proposed ANMPD method for SIFT. Table 1 reports the minimum clustering error rate (CER) obtained for the different values of N. It shows that the CER decreases and then increases with a minimum value at N=5. In next experiments, we fixe N to 5.

The second experiment is done in order to study the impact of keyface selection. Results show that the arbitrary choice of the middle face gives a CER equal to 43.7%. However, the proposed method for selecting keyfaces gives a CER equal to 28.4%.

The third experiment is done in order to compare the ANMPD with Lowe's matching and Minimum pair distance matching. Results in Table 2 show that our proposed method outperforms the Lowe's matching by an absolute gain of 7.5%, the MPD method by 26.7% and the MPD on regular grid by 3%.

The fourth experiment is done in order to compare the clustering using each descriptor alone. Table 3 shows that the descriptor that gives best results is the 3D-Histogram of the clothing with a $CER = 16.8\%$. The second good results are provided by SIFT matching with $CER = 28.4\%$. The two other descriptors are consecutively the clothing texture and the skin color of the face. The corresponding stopping criteria for each descriptor are also reported. These thresholds are used in equation (11) and (17) to compute similarity matrices for the hierarchical clustering.

The fifth experiment is done in order to report the behavior of the proposed fusion method compared to the four descriptors at different levels of the clustering process. Figure 9 shows that the proposed clustering is better than almost all descriptors each one taken alone.

For example:

- when the number of clusters is equal to 400, the proposed clustering outperforms the best one (skin color descriptor) by an absolute gain of 1.7%.
- when the number of clusters is equal to 250, the CER of the proposed method is equal to 44.8% however the best of the four descriptors was the SIFT with CER equal to 46.4%.
- when the number of clusters is equal to 25, the CER of the proposed method is equal to 14.5% however the best of the descriptor was the 3D-histogram of the clothing with CER equal 42.3%.
- the best CER value is 13%. It is obtained for a number of clusters equal to 50.

The sixth experiment is done in order to evaluate the impact of using the dominant color. Figure 10 shows that at the beginning of the clustering process (number of cluster higher than 200), no real comparison can be made. However, when the number of clusters approaches the real number of people, the impact of using the dominant color is highest: when the number of clusters is equal to 50, the absolute gain is 34.9%.

6. CONCLUSION

In this paper, a people clustering system is reconstructed using both face and clothing information. After extracting the *keyface* from each video sequence, a method for face matching is proposed based on SIFT features using Lowe's algorithm and ANMPD distance. Then, a method for matching clothing using 3D histograms and dominant color is proposed. Finally, a 3-levels hierarchical bottom-up clustering algorithm that combines SIFT features, skin color, 3D histograms and clothing texture is described. Experiments done on a *talk show* TV program containing about 520 tracks show the efficiency of our proposed clustering system. As future work, the audio component [6] will be combined with the described visual component in order to build an audiovisual people clustering system.

[2] http://www.itl.nist.gov/iaui/894.01/

Table 1: Clustering Error Rate for different N values used in Equation (6).

N	1	2	3	4	**5**	6	7	8
CER(%)	55.1	49.1	32.8	31.2	**28.4**	30.2	33	35.1

Table 2: Comparison between different sift matching techniques: Lowe's matching, MPD matching, MPD matching on regular grid and the proposed ANMPD matching.

	Lowe's matching	MPD	MPD on regular grid	proposed ANMPD
CER(%)	35.9	55.1	31.4	28.4

Table 3: Minimum clustering error rate for each visual descriptor: 3D-Histogram of the clothing, texture of the clothing, skin color of the face, and sift features of the face. The thresholds that corresponds to the stopping criterion are also reported

	SIFT	Skin	Hist	Texture	proposed clustering
CER (%)	28.4	56.6	16.8	55.5	13.0
Stopping criterion	$Thr1 = 0.41$	$Thr2 = 3.2$	$Thr3 = 3.3$	$Thr4 = 0.126$	-

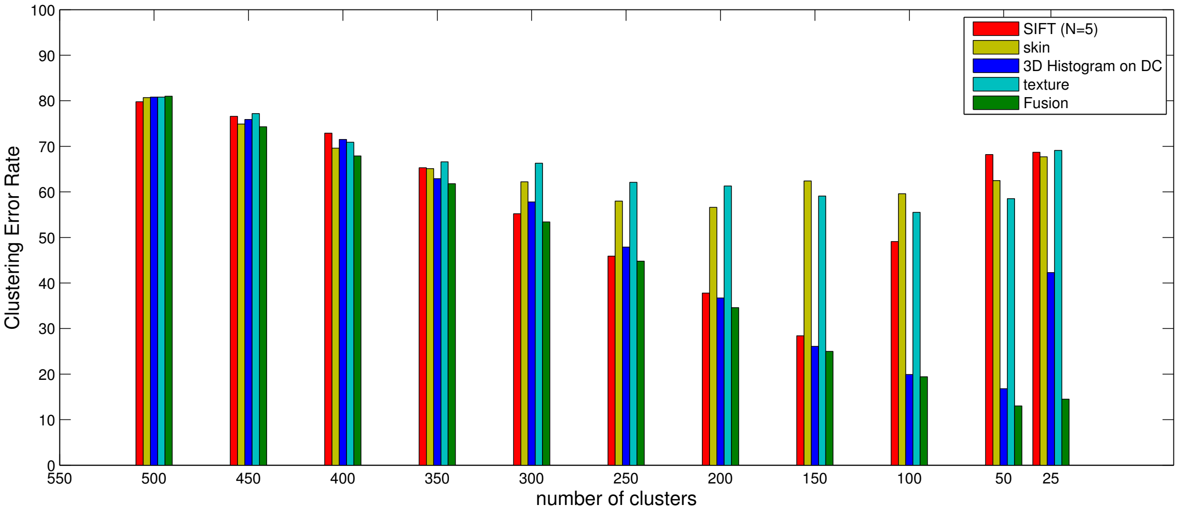

Figure 9: Comparison between the four features and the proposed clustering method.

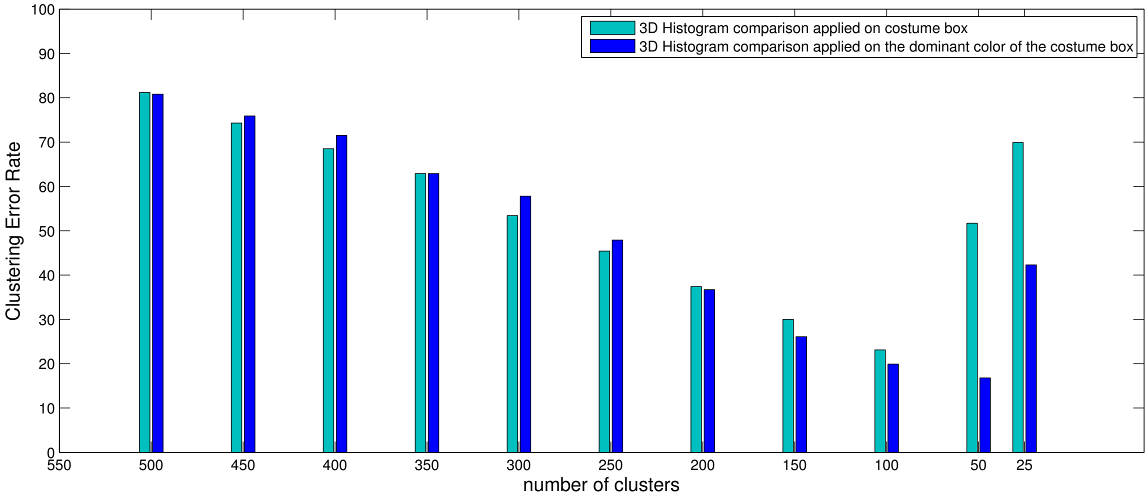

Figure 10: Comparison between applying the Histogram comparison directly on the costume box and applying it on the dominant color area.

7. REFERENCES

[1] O. Arandjelovic and A. Zisserman. Automatic face recognition for film character retrieval in feature-length films. In *CVPR'05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 860–867. IEEE Computer Society, 2005.

[2] J. S. Beis and D. G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proc. IEEE Conf. Comp. Vision Patt. Recog*, pages 1000–1006, 1997.

[3] P. N. Belhumeur, J. P. Hespanha, J. ao P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:711–720, 1996.

[4] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli. On the use of sift features for face authentication. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06.*, pages 35–41, June 2006.

[5] S.-F. Chang, J. He, Y.-G. Jiang, E. El Khoury, C.-W. Ngo, A. Yanagawa, and E. Zavesky. Columbia University/VIREO-CityU/IRIT TRECVID2008 High-Level Feature Extraction and Interactive Video Search. In *TREC Video Retrieval Workshop (TRECVID), NIST in Gaithersburg, MD.* NIST, 2008.

[6] E. El-Khoury, C. Senac, and J. Pinquier. Improved speaker diarization system for meetings. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 4097–4100, 2009.

[7] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy – automatic naming of characters in tv video. In *Proceedings of the British Machine Vision Conference, BMVC'06*, page III:899, 2006.

[8] A. W. Fitzgibbon and A. Zisserman. On affine invariant clustering and automatic cast listing in movies. In *ECCV'02: Proceedings of the 7th European Conference on Computer Vision-Part III*, pages 304–320, London, UK, 2002. Springer-Verlag.

[9] S. Haidar, P. Joly, and B. Chebaro. Mining for video production invariants to measure style similarity. *International Journal of Intelligent Systems (IJIS)*, 21(7):747–763, july 2006.

[10] M. hsuan Yang. *Encyclopedia of Biometrics*, chapter : Face Detection. Springer, July 2009.

[11] G. Jaffré and P. Joly. Costume: a New Feature for Automatic Video Content Indexing. In *RIAO'2004 : Coupling approaches, coupling media and coupling languages for information retrieval, Avignon*, pages 314–325. C.I.D., 2004.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[13] S. Maji and R. Bajcsy. Fast unsupervised alignment of video and text for indexing/names and faces. In *MS'07: Workshop on multimedia information retrieval on The many faces of multimedia semantics*, pages 57–64. ACM, 2007.

[14] B. S. Manjunath and W. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) - Special issue on Digital Libraries*, 18(8):837–42, Aug 1996.

[15] A. Martinez and R. Benavente. The ar face database. Technical report, CVC Technical Report, 1998.

[16] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *WACV'94*, pages 138–142, 1994.

[17] A. F. Smeaton, P. Over, and A. R. Doherty. Video shot boundary detection: Seven years of trecvid activity. *Computer Vision and Image Understanding*, 2009.

[18] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. In *Proc. Graphicon-2003*, pages 85–92, 2003.