Towards an Automatic Face Indexing System for Actor-based Video Services in an IPTV Environment

Jae Young Choi, Wesley De Neve, and Yong Man Ro, Senior Member, IEEE

Abstract — Automatic face indexing is an important technique for realizing actor-based video services in an IPTV environment. This paper proposes a novel face indexing system that takes advantages of the internet connection of an STB to construct a FR engine that is equipped with a high number of training face images. In addition, we use a face clustering technique to obtain multiple face images of the same subject from a sequence of video frames. The clustered face images are combined using a weighted feature fusion scheme, resulting in a considerable enhancement in face indexing accuracy. The effectiveness of the proposed system is validated using more than 300,000 video frames, collected from five video clips containing drama or movie content. The experimental results show that the proposed system can achieve a face annotation accuracy that is feasible for practical applications.

Index Terms — Internet Protocol Television (IPTV), Face indexing, Face recognition, Digital video content

I. INTRODUCTION

Recently, the number of Internet Protocol Television (IPTV) deployments has grown rapidly, offering various digital video services. In an IPTV environment, typically owned and managed by a telecom provider, digital video content is delivered to consumers over a broadband connection using the Internet Protocol (IP) [1]. That way, IPTV allows for the convergence of digital television broadcasting and Web-based services. A core element of an IPTV environment is the set-top box (STB), acting as the gateway between the IP network of the telecom provider and the home network and digital television of an end-user. Nowadays, STBs often come with a high storage capacity, making these devices suitable for the storage and management of a personal media library.

The amount of video content that can be consumed through IPTV-based services is increasing at a high pace. As such, personalization is important in order to enable effective consumption of video content. Indeed, personalization empowers users to consume video content based on their own preferences (e.g., favorite genre or favorite actors). To facilitate user-centered video services in IPTV systems, content-based indexing (or annotation) techniques are strongly needed [2]-[3]. In particular, recent user studies have shown that users prefer to index, search, and organize their video content based on “who” is appearing in the content [4]. Further, as dramas and movies typically consist of scenes involving one or more well-known actors [5], actor-based video indexing for IPTV-based video services may have a high commercialization potential.

To date, the scientific literature only contains a few papers that deal with automatic face indexing in digital video content. In [6], the authors propose an interactive querying method for indexing the names of actors in video content. In this method, users are first asked to click on actors in a video frame. Face detection is then applied to the clicked video frame in order to determine the location of faces. Subsequently, a frame buffering technique (relying on object tracking) is used to select the best frontal face from a pre-specified number of frames before and after the clicked frame. Finally, a selected face is fed into a Gabor filter-based FR step. In [7], during the training stage, a video sequence is split into a number of subgroups by using both temporal and visual information. The resulting subgroups are subsequently merged into a set of groups, where each group corresponds to the identity of a particular subject (that is to be indexed). The identity of a query face is determined by comparing the distance computed between a query face and a representative face sample selected from each group. In [8], the authors propose a variant of PCA, so-called self-eigenfaces, for use in video indexing applications. Multiple sets of eigenfaces are constructed for each target subject using face images obtained for different views. Each query face is separately projected onto the individual eigenfaces and reconstructed for FR purposes.

As previously mentioned, a limited number of techniques have thus far been proposed for indexing faces in video content. Moreover, to our best knowledge, an effective face indexing system for facilitating actor-based video services in an IPTV environment has not been described yet in the scientific literature. In this paper, we propose a novel face indexing system that is particularly effective for IPTV systems. Moreover, the proposed system is able to attain high face annotation accuracy for digital video content. Given a video sequence, our face indexing system aims to automatically find a set of video shots that show actors of interest. Compared to previous work, the main technical contributions of our work are as follows:

- Using the Internet connection of an STB and metadata available in an Electronic Program Guide (EPG), a high number of training face images for actors of interest are collected from the Internet. This allows building a FR engine that is able to reliably recognize actors of interest in dramas or movies.

The authors are with the Image and Video Systems Lab, Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea (email: jygchoi@kaist.ac.kr, wesley.deneve@kaist.ac.kr, and ymro@ee.kaist.ac.kr).

Manuscript received January 10, 2010 0098-3063/10/$20.00 © 2010 IEEE
To take advantage of the temporal dimension of video content, we propose an effective face clustering technique and a novel FR method using weighted feature fusion. Thanks to the use of face clustering, face images of the same subject can be obtained from a sequence of video frames. Multiple face features (for a particular subject), extracted from the clustered face images, are then combined using the proposed weighted feature fusion solution, improving face annotation accuracy.

In general, face images detected in video frames may have a lower visual quality due to motion blurring and the use of compressed video formats (such as MPEG-2 Video). To cope with this problem, we make effective use of color face features [9] in order to build a face indexing system that is robust to face images with a low visual quality.

The efficacy of the proposed system is validated using more than 300,000 video frames collected from five video clips containing drama or movie content. The experimental results show that the proposed system can achieve a face annotation accuracy that is feasible for practical applications. Further, the proposed system is straightforward to implement and computationally inexpensive.

The rest of this paper is organized as follows. In Section II, we present the overall architecture of our face indexing system. The construction of a FR engine using the Web is described in Section III. The proposed video FR technique is outlined in Section IV. Section V presents experimental results, validating our face indexing system in terms of both face annotation accuracy and run-time performance. Conclusions and directions for future research are presented in Section VI.

II. OVERALL SYSTEM ARCHITECTURE

Computer-based face detection (FD) and face recognition (FR) are key technologies in order to realize an automatic system for indexing faces of actors. While automatic FD has become a mature technique [10], automatic FR is still far from being adequate for many practical applications. This can mostly be attributed to the fact that face images, detected in video frames, are often the subject of severe changes in illumination, pose, and resolution (see Fig. 4 in Section V). For this reason, this paper will focus on the problem of video face recognition (rather than the problem of face detection) in order to effectively index the actors in video content.

Fig. 1 shows the overall architecture of the proposed face indexing system, running on the STB of a particular user. The STB contains a personal media library consisting of drama and movie content (e.g., prerecorded using the STB, downloaded from the Internet, or personal DVD copies).

As outlined in Fig. 1, the proposed face indexing system mainly consists of two parts: 1) construction of a FR engine using the Web and 2) video face recognition. Using the Internet connection of an STB, actor names are first retrieved from an online drama and movie information provider. When prerecorded video content is consumed, EPG metadata can be used as well in order to retrieve actor names without requiring any user interaction. Training and target face images for actors that need to be annotated (this is, the actors of interest or the so-called target actors) are subsequently acquired from the Web. The constructed FR engine and a set of target face images are then passed to the video FR stage. In general, video shots are the primitives for content-based operations such as indexing and retrieval. The frames present in each video shot are processed by the face clustering step. The goal of face clustering is to group face images (detected in a sequence of video frames within a shot) into a representative set of clusters, each corresponding to a unique subject. Multiple face images present in each face cluster are
transformed into corresponding color face features. Multiple color face features of the same subject are then combined using the proposed method for weighted feature fusion. Finally, based on the obtained FR results, face images within a face cluster are annotated with the identity of an actor of interest. A more detailed description of our face indexing system is given in the following sections.

III. CONSTRUCTION OF A FR ENGINE USING THE WEB

In pattern recognition theory, and in particular in the area of supervised learning, a classifier becomes in general more effective when more training samples can be used to learn the dominant characteristics of each class [11]. Consequently, the availability of a high number of training face images for target subjects is advantageous when having to build a reliable FR engine [20]-[21].

Starting from the observation discussed in the previous paragraph, the aim of using the Web during the construction of a FR engine is to acquire a high number of training face images for the actors of interest. To this end, the user first enters the title of a drama or movie in the user interface provided by the STB. The STB then communicates the title to a drama or movie information service provider. This service provider returns a list of names of actors appearing in the drama or movie content to the STB. In a next step, the STB uses the list of actor names to perform an image search on the Web. In other words, the actor names are used to collect a high number of training face images through a Web image search performed by the STB.

Note that the list with actor names can also be used to restrict the set of likely matches for a given query face to a limited number of actors (usually less than a dozen of actors). This makes it possible to attain high annotation accuracy when targeting face indexing in drama or movie content. Indeed, it is well understood that FR tasks become less effective as the number of target subjects stored in the target database (DB) increases (e.g., when more than hundreds of target subjects are present in the target DB) [12].

After having performed an image search on the Web, face images are automatically detected and extracted from the collected images using state-of-the-art FD techniques (e.g., by making use of the FD technique described in [10]). The detected face images are then divided into two sets: a training DB and a target DB. The training images are used to build a FR engine that is able to recognize actors of interest. Specifically, in a typical FR system, building a FR engine mainly consists of constructing a face feature extractor [13]. In our research, the training images are used to create a face feature extractor that is able to extract color face features (as will be explained in Section IV-B).

IV. VIDEO FACE RECOGNITION

In this section, we discuss the proposed video FR method in more detail. In order to decompose a video sequence into a number of shots, the technique proposed in [14] was used. This technique, which has proven to be effective and accurate, detects video shots by making use of a color histogram (computed for each video frame). The remaining modules (as shown in Fig. 1), needed to perform video FR, are discussed in the following subsections.

A. Face Clustering

The aim of face clustering is twofold: first, all facial images of the same subject should be grouped into a single cluster and second, the face images of different subjects should be part of different clusters.

We now describe our face clustering method in more detail. Let \( I_i \) be the \( i \)-th face image detected in a video frame that belongs to a particular shot, where \( 1 \leq i \leq L \) and where \( L \) denotes the total number of face images detected in all video frames within the shot in question. Note that face images can be extracted using any state-of-the-art face detection technique. The face feature of \( I_i \) is denoted as \( f_i \). Note that \( f_i \) can be obtained using any state-of-the-art face feature extraction technique (e.g., using global or local face features [15]).

Since we have no a priori information about the possible identity of subjects or the nature of face feature samples, we make use of an unsupervised clustering technique for clustering face images. Specifically, we have adopted average-linkage based hierarchical agglomerative clustering (HAC) [16] in our research for the purpose of face clustering. The HAC-based face clustering algorithm is summarized below.

(a) Assuming that \( L \) face features are available in a single shot, HAC starts with \( L \) singleton face clusters. The singleton face clusters are denoted as \( C_i \), where \( i = 1, \ldots, L \). Note that each \( C_i \) consists of a single face feature \( f_i \) during the initial stage.

(b) Calculate the average dissimilarity between \( C_i \) and \( C_j \) by summing the pairwise dissimilarities between the face features in the two different face clusters:

\[
D_{\text{cluster}}(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{f_{im} \in C_i} \sum_{f_{jn} \in C_j} D_{\text{face}}(f_{im}, f_{jn}),
\]

where \(|\cdot|\) denote the cardinality of a set, \( D_{\text{face}}(\cdot) \) is a function that measures the dissimilarity (e.g., distance) between a pair of face features, \( m \neq n \), and \( 1 \leq m, n \leq L \).

(c) Find the nearest two face clusters \( C_i \) and \( C_j \) by comparing all \( D_{\text{cluster}}(C_i, C_j) \) one by one in the following way:

\[
(C_i, C_j) = \arg \min_{i \neq j} D_{\text{cluster}}(C_i, C_j),
\]

(d) Merge the two nearest clusters into a single cluster, denoted as \( C_{ij} = C_i \cup C_j \), and remove \( C_i \) and \( C_j \).
(e) If the value of $D_{	ext{cluster}}(\mathbf{C}_i, \mathbf{C}_j)$ is lower than a pre-determined stopping threshold $\xi$, then repeat steps (b), (c), and (d), else terminate the face clustering process.

For determining an optimal $\xi$, we employ the stopping threshold selection method proposed in [16] in our HAC-based face clustering strategy.

Fig. 2 visualizes HAC-based face clustering. In Fig. 2, face images are first detected in a sequence of video frames within a shot. We assume that the total number of detected face images is eight. As such, the number of face clusters in the initial stage is eight (i.e., $L = 8$). In the final iteration, four face images belonging to the same subject are grouped into $\mathbf{C}_1$, while four face images belonging to another subject are assigned to $\mathbf{C}_2$.

B. Extraction of Color Face Features

For the sake of simplicity and without any loss of generality, we further denote a particular face cluster as $\mathbf{C}_i$. Let $\mathbf{I}_i^{(n)}$ and $\mathbf{I}_i^{(m)}$ be the $m^{th}$ query face image (this is, a face image that needs to be recognized) and the $n^{th}$ target face image, respectively, where $1 \leq m \leq |\mathbf{C}_i|$, $1 \leq n \leq G$, and where $G$ is the total number of target subjects (i.e., the actors of interest) that are enrolled in the target database. Note that $\mathbf{I}_i^{(n)}$ and $\mathbf{I}_i^{(m)}$ are assumed to be red-green-blue (RGB) color face images.

In order to extract the color face features of $\mathbf{I}_i^{(n)}$ and $\mathbf{I}_i^{(m)}$, $\mathbf{I}_i^{(n)}$ and $\mathbf{I}_i^{(m)}$ are first converted into images having a different color representation (e.g., the RGB images are converted into YIQ or HSV images). Let us assume that a total of $K$ different color component images are generated (e.g., luminance ($Y$) or chrominance component ($C_b$ or $C_r$) images in the $YC_bC_r$ color space). We denote the $k^{th}$ color component vectors [9] of $\mathbf{I}_i^{(n)}$ and $\mathbf{I}_i^{(m)}$ as $\mathbf{s}_i^{(n,k)}$ and $\mathbf{s}_i^{(m,k)}$, respectively, where $k = 1, \ldots, K$.

Note that each color component vector is a column vector. This column vector is the result of row-wise scanning of the pixel elements of a two-dimensional color component image.

Let us denote the $k^{th}$ face feature extractor as $\varphi_k$ (see [13] for a detailed discussion of face feature extractors). Further, let us assume that $\varphi_k$ is formed with a training set consisting of the $k^{th}$ color component vectors, where the latter are generated from the training images associated with the target subjects [9]. Then, the low-dimensional features of the $k^{th}$ color component vectors $\mathbf{s}_i^{(n,k)}$ and $\mathbf{s}_i^{(m,k)}$ are obtained as follows (using the corresponding $\varphi_k$):

$$
\mathbf{f}_i^{(n,k)} = \varphi_k(\mathbf{s}_i^{(n,k)}) \quad \text{and} \quad \mathbf{f}_i^{(m,k)} = \varphi_k(\mathbf{s}_i^{(m,k)}),
$$

where $\mathbf{f}_i^{(n,k)}, \mathbf{f}_i^{(m,k)} \in \mathbb{R}^{K}$ and $k = 1, \ldots, K$.

In order to generate the color face features of $\mathbf{I}_i^{(n)}$ and $\mathbf{I}_i^{(m)}$, $K$ complementary color component features are combined at the level of the features (by concatenating the color component features in column order):

$$
\mathbf{f}_i^{(n)} = \left[ \left( \mathbf{f}_i^{(n,1)} \right)^T \left( \mathbf{f}_i^{(n,2)} \right)^T \ldots \left( \mathbf{f}_i^{(n,K)} \right)^T \right]^T \quad \text{and} \quad \mathbf{f}_i^{(m)} = \left[ \left( \mathbf{f}_i^{(m,1)} \right)^T \left( \mathbf{f}_i^{(m,2)} \right)^T \ldots \left( \mathbf{f}_i^{(m,K)} \right)^T \right]^T,
$$

Fig. 3. Color face feature extraction process for query and target face images. The text enclosed in brackets (placed to the right-hand side of each face image) indicates the grayscale or color format of the corresponding face images. The $YIQ$ color space is used as an example.
where $T$ denotes the matrix transpose operator, $f_q^{(m)}, f_i^{(n)} \in \mathbb{R}^k$, and $F = \sum_{i=1}^{k} F_i$. Note that, prior to their concatenation, $f_q^{(m,i)}$ and $f_i^{(n)}$ need to be individually normalized in order to have zero mean and unit variance. Fig. 3 shows the color face feature extraction process for $I_q^{(m)}$ and $I_i^{(n)}$.

Subsequently, $f_q^{(m)} (m = 1, \ldots, |C|)$ and $f_i^{(n)} (n = 1, \ldots, G)$ are the subject of weighted feature fusion in order to annotate the face images ($I^{(m)}$) that are available in $C$. This process is explained in more detail in the following subsection.

C. Face Recognition Using Weighted Color Feature Fusion

This section explains a FR technique that makes use of weighted feature fusion in order to merge multiple color face features. This approach allows dealing with several defective face images in a face cluster. Defective face images may be the present in a face cluster due to variations in terms of viewpoint, illumination, and compression artifacts. We regard defective face images as outliers, whereas a prototype image can be considered representative for a single face cluster.

The impact of outliers on the effectiveness of face recognition should be minimized. To this end, we compute a weight for each individual $f_q^{(m)}$, and where this weight is representative for the distance between $f_q^{(m)}$ and a corresponding prototype image. Further, it is well-known that the median is more resilient to outliers than the mean. Hence, we adopt a median feature vector, denoted as $\tilde{f}_q$, as a prototype for a face cluster. Note that each element of $\tilde{f}_q$ is filled with the median value of all corresponding elements of $f_q^{(m)} (m = 1, \ldots, |C|)$. To diminish the influence of elements of $f_q^{(m)}$ that are far away from $\tilde{f}_q$, a penalty-based Minkowski distance is defined as follows:

$$d_m = \left( \sum_{k} \left[ \Omega(\|\Gamma_k(\tilde{f}_q) - \Gamma_k(f_q^{(m)})\|) \right]^{\frac{1}{p}} \right)^{1/p},$$

where

$$\Omega(|x|) = \begin{cases} \tau |x| & \text{if } |x| > \delta \cdot \sigma_k \\ |x| & \text{otherwise} \end{cases},$$

and $\Gamma_k(\cdot)$ is a function that returns the $k^{th}$ element of the argument vector, $\sigma_k$ stands for the standard deviation computed over the $k^{th}$ element samples of the feature vectors $f_q^{(m)}$ that are part of $C$, and $\delta$ and $\tau$ denote a user-specific threshold and a penalty constant, respectively. Based on our experimental results, 2.2 and 2.0 have been found to be reasonable values for $\delta$ and $\tau$, respectively. It should be emphasized that in (5), the distance $d_m$ is forced to increase if the difference between an element in $\tilde{f}_q$ and an element in $f_q^{(m)}$ exceeds a certain $\delta \cdot \sigma_k$. The increase in distance is controlled by the parameter $\tau$ (see (5)).

Using $d_m$, as given by (4), and a soft-max function [11], we compute a weight in order to adjust the influence of $f_q^{(m)}$ on the fusion of face features:

$$w_m = \frac{\exp(-d_m)}{\sum_{g} \exp(-d_g)}.$$  

Note that $d_m$ should be normalized to have zero mean and unit standard deviation prior to the computation of $w_m$. In this paper, the widely used ‘z-score’ technique is employed to normalize the distance scores. Other distance score normalization techniques are explained in detail in [17].

Using $w_m$, a single feature vector can be computed as a weighted average of the individual feature vectors $f_q^{(m)}$ in $C$ using the following equation:

$$f_q = \sum_{m} w_m \cdot f_q^{(m)}.$$  

In (7), by assigning a higher weight to the reliable face features and a lower weight to the other face features (i.e., the outliers), the chance of assigning outliers to the wrong subject class can be reduced.

To perform face annotation on $I_q^{(m)}$, a nearest neighbor (NN) classifier is applied to determine the identity of $I_q^{(m)}$. This is done by finding the smallest distance between $f_q^{(m)}$ and $f_i^{(n)} (n = 1, \ldots, G)$, as follows:

$$l(I_q^{(m)}) = l(I_i^{(m)}) \text{ and } n^* = \arg\min_{g \in \mathbb{G}} D_{\text{norm}}(f_q, f_i^{(n)}),$$

where $l(\cdot)$ is a function that returns the identity of an input face image. Using (8), all $I^{(m)}$ contained in $C$ are annotated with subject identity $l(I_i^{(m)})$ in a batch manner.

V. EXPERIMENTS

In order to investigate the effectiveness and the efficiency of the proposed face indexing system, a number of experiments were performed, using five video clips containing drama or movie content. Table I discusses a number of characteristics of the video clips used in our experiments. Note that the resolution of the video clips (described in Table I) is 720x480 pixels for 4:3 Standard Definition (SD) video.
Fig. 4. Example video frames used in our experiments. Each row shows example frames from four selected video clips described in Table I. Note that the detected face images (that are part of the ground truth) of a number of target actors are placed to the bottom side of each corresponding video frame. All of the detected face images were individually rotated and rescaled to 150 \times 130 pixels so that eye centers were placed on fixed pixel locations (as recommended by the FERET protocol [19]).

TABLE I

<table>
<thead>
<tr>
<th>Video clip ID</th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>Drama</td>
<td>Movie</td>
<td>Movie</td>
<td>Drama</td>
<td>Movie</td>
</tr>
<tr>
<td>Duration</td>
<td>23′17″</td>
<td>69′39″</td>
<td>46′58″</td>
<td>42′17″</td>
<td>40′93″</td>
</tr>
<tr>
<td>No. of frames</td>
<td>33,366</td>
<td>99,917</td>
<td>67,080</td>
<td>60,726</td>
<td>58,938</td>
</tr>
<tr>
<td>No. of shots</td>
<td>294</td>
<td>828</td>
<td>822</td>
<td>537</td>
<td>498</td>
</tr>
<tr>
<td>Frame rate</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
</tbody>
</table>

Given a particular video clip, we constructed a corresponding ground truth composed of manually labeled shots as well as manually labeled face images. Recall that our face indexing system aims at detecting whether or not a particular actor appears in a given shot of a given video clip. In line with this goal, principal cast members for each video clip were designated as the actors of interest. Then, labels identifying these target actors were manually assigned to the shots (that contain the pre-specified target actors) of each video clip. In addition, Viola-Jones’ face detection algorithm [18] was applied to all frames in each video clip in order to facilitate the manual labeling of face images. These manually labeled face images were used to evaluate the performance of the face clustering technique outlined in Section IV-A.

Table II describes the five constructed ground truth (GT) datasets. In addition, Fig. 4 shows a number of example video frames, as well as the detected face images used in our experiments. The detected face images are significantly challenging for current FR techniques due to variations in pose and illumination, a low visual quality, and even occlusion.

TABLE II

<table>
<thead>
<tr>
<th>Video clip ID</th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of target actors to be annotated</td>
<td>13</td>
<td>10</td>
<td>7</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>No. of shots containing target actors</td>
<td>198</td>
<td>654</td>
<td>567</td>
<td>341</td>
<td>265</td>
</tr>
<tr>
<td>No. of frames containing target actors</td>
<td>18,211</td>
<td>71,345</td>
<td>41,112</td>
<td>48,427</td>
<td>27,768</td>
</tr>
</tbody>
</table>

In an actor-based video service scenario, it is reasonable to assume that a FR engine has been customized to recognize a specific group of actors that are of interest to the user. Thus, face feature extraction methods using supervised techniques [20] may be suitable for building a FR engine in our face indexing system. The underlying reason for this is that face feature extraction methods relying on supervised techniques are usually characterized by a high FR performance when a high number of training samples per target subject is available and when the number of target subjects is low. In our experiments, Regularized Linear Discriminant Analysis (RLDA) [21] was adopted for the purpose of face feature extraction. In [21], RLDA shows considerably better FR performance than other popular face feature extraction methods that make use of supervised techniques. The Euclidean distance was used to measure dissimilarity (i.e., to measure $D_{\text{face}}(\cdot)$, as shown in (8)).
As discussed in Section III, the training face images for each target actor are obtained from the Web by means of the Internet connection of an STB. Note that more than 90% of the retrieved Web images were found to correctly contain the predefined target actors. Noisy Web images (i.e., Web images containing subjects that are different from the target actors) were removed from the set of acquired training face images. Hence, in our experiments, an average of 10 training face images was used for each target actor in order to form a face feature extractor. It is worth noting that, when referring to the literature in the area of face recognition [12], [20]-[21], the use of more than eight training images for a particular target subject is in general sufficient to build a reliable face feature extractor.

For the purpose of face clustering, the popular Local Binary Pattern (LBP) [22] was adopted as face feature. Note that the final goal of face clustering is to correctly group face images that belong to the same actor. For this reason, we focus on assessing the accuracy of merging face images by means of unsupervised face clustering, rather than assessing the accuracy of shot boundary detection.

The $FScore$ metric [23] was adopted to quantify the clustering performance. As described in [23], an $FScore$ value may vary from 0 to 1. This means that the $FScore$ will be one when every target actor has a corresponding face cluster that contains all face images belonging to the target actor in question. Hence, the higher the $FScore$ value, the better the clustering result from the point-of-view of natural grouping. In addition, to evaluate the face annotation performance at the level of shots, the following precision and recall metrics were used:

\[
\text{precision} = \frac{1}{G} \sum_{n=1}^{G} \frac{N_{\text{correct}}^{(n)}}{N_{\text{retrieval}}^{(n)}} \quad \text{and} \quad \text{recall} = \frac{1}{G} \sum_{n=1}^{G} \frac{N_{\text{correct}}^{(n)}}{N_{\ground}^{(n)}}, \tag{9}
\]

where $G$ is the total number of target actors (this, the number of actors that needs to be annotated), $N_{\text{retrieval}}^{(n)}$ is the number of retrieved shots annotated with an identity label $n$, $N_{\text{correct}}^{(n)}$ is the number of shots correctly annotated with identity label $n$, and $N_{\ground}^{(n)}$ is the number of shots annotated with identity label $n$ in the ground truth described in Table II.

In order to show the effectiveness of face clustering, $FScore$ values are presented in Table III, for each video clip used. Note that all $FScore$ values listed in Table III denote values that have been averaged over all of the shots automatically detected for each video clip. In addition, the average $FScore$ values have been computed for an optimal stopping threshold (determined using the technique proposed in [16]). Looking into the results in Table III, the $FScore$ values are higher than 0.89 for all video clips used. Based on the fact that the $FScore$ becomes one when perfect clustering results are achieved, we can verify that the proposed face clustering method is able to attain a reliable clustering performance (for an optimal stopping threshold).

![Fig. 5. Distribution of the number of face clusters having correctly clustered face images (all belonging to the same subject).](image-url)

Fig. 5 shows the distribution of the number of face clusters having correctly clustered face images (belonging to the same subject). In Fig. 5, the distribution is determined by calculating a histogram, and where a bin represents the fraction of face clusters having a particular number of correctly clustered face images. This histogram is generated using a total of 8,935 clusters. These clusters are the result of performing face clustering on the 2,952 shots included in the GT (as described in Table II). Further, the histogram has been normalized by dividing each of its components by the total number of face clusters (i.e., 8,935 face clusters). From Fig. 5, we can see that as much as 94% of the face clusters contain more than five correctly clustered face images that all belong to the same subject. This result indicates that our FR technique, which makes use of weighted feature fusion, is performed on at least five correctly clustered face images in most of the face clusters.

For comparison purposes, precision and recall values were obtained for FR techniques focusing on still images [13]. The resulting performance is referred to as the baseline annotation performance. Current FR techniques focusing on still images can be largely divided into two approaches: global and local appearance-based FR [15]. Both approaches were used in our experiments. For baseline FR relying on the global appearance-based method, RLDA (mentioned above) was adopted for the purpose of face feature extraction. Note that the training set used to form an RLDA-based feature

<table>
<thead>
<tr>
<th>Table III</th>
<th>$FScore$ Values Averaged Over All of the Video Shots Generated for Each Video Clip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video clip ID</td>
<td>Average $FScore$</td>
</tr>
<tr>
<td>V1</td>
<td>0.92</td>
</tr>
<tr>
<td>V2</td>
<td>0.94</td>
</tr>
<tr>
<td>V3</td>
<td>0.91</td>
</tr>
<tr>
<td>V4</td>
<td>0.89</td>
</tr>
<tr>
<td>V5</td>
<td>0.96</td>
</tr>
</tbody>
</table>
Fig. 6. Comparison of the face annotation performance obtained for the proposed method and two baseline methods: (a) precision and (b) recall. Note that the proposed method performs weighted color feature fusion on grouped face images. The grouped face images are the result of face clustering using an optimal stopping threshold.

We have measured the execution time required to annotate video clip ‘V2’ (containing 654 manually labeled shots, 71,345 frames, and having a duration of about 69 minutes) on a PC with an Intel Pentium IV 2.4 GHz CPU processor. The time needed to execute shot segmentation in conjunction with face clustering and weighted color feature fusion is about 214 seconds (around 320 ms per shot). The time needed for selecting an optimal stopping threshold value during face clustering has also been included in the total execution time. On contrary, the processing time required to create a FR engine (including the collection of training images through a Web-based image search and the training of a face feature extractor) is not considered: the construction of a FR engine is performed in an off-line manner in our face indexing system.

VI. CONCLUSION

In this paper, we have presented a face indexing system that aims at deployment in an IPTV environment. The proposed system allows finding a set of shots in drama or movie clips that contain actors that are of interest to the user. Our face indexing system was implemented and tested using challenging video content, containing face images with severe variations in terms of pose, illumination, and visual quality. The experimental results demonstrate that our face indexing system is able to accurately annotate faces of actors in drama and movie video content. Further, all operations related to video FR are unsupervised and have a low computational complexity. Further, we also measured the runtime performance of the proposed face indexing system in a PC environment. We observed that our face indexing system needs about 320 ms in order to process a video shot. We believe that this performance is close to the performance of a real set-top box environment. Current set-top boxes come with support for high-speed floating point units and matrix operations. Consequently, we believe that our face indexing system enables users of IPTV systems to consume video clips that contain their favorite actors.

In this paper, we assume that a FR engine is constructed in an off-line manner. However, user preferences may change over time. Therefore, it should be possible to dynamically
update a FR engine by retraining its associated face feature extractor, using face images of new target actors. Off-line learning methods for the construction of a FR engine often come with high memory consumption and high computational cost, making retraining computationally expensive. Hence, future research will focus on the construction of a FR engine using online-learning [24], which allows dynamically adjusting the FR engine to evolving user preferences.

REFERENCES


BIOGRAPHIES

Jae Young Choi received the B.S. degree from Kwangwoon University, Seoul, South Korea, in 2004 and the M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2008. He is currently working toward the Ph.D. degree at KAIST. In 2007, he worked as an intern researcher for the Electronics and Telecommunications Research Institute (ETRI) in South Korea. In 2008, he was a visiting student researcher at the University of Toronto in Canada. His research interests include face recognition/detection, image/video indexing, pattern recognition, machine learning, the Social Web, and personalized broadcasting technologies. He is a student member of the IEEE.

Wesley De Neve received the M.Sc. degree in Computer Science and the Ph.D. degree in Computer Science Engineering from Ghent University, Ghent, Belgium, in 2002 and 2007, respectively. He is currently working as a senior researcher for the Image and Video Systems Lab (IVY Lab) in the position of assistant research professor. IVY Lab is part of the Department of Electrical Engineering of the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. Prior to joining KAIST, he was a post-doctoral researcher at both Ghent University - IBBT in Belgium and the Information and Communications University (ICU) in South Korea. His research interests and areas of publication include the coding, annotation, and adaptation of image and video content, GPU-based video processing, efficient X-XML processing, and the Semantic and the Social Web.

Yong Man Ro (M’92-SM’98) received the B.S. degree from Yonsei University, Seoul, South Korea, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. In 1987, he was a visiting researcher at Columbia University, and from 1992 to 1995, he was a visiting researcher at the University of California, Irvine and KAIST. He was a research fellow at the University of California, Berkeley and a visiting professor at the University of Toronto in 1996 and 2007, respectively. He is currently holding the position of full professor at KAIST, where he is directing the Image and Video Systems Lab. He participated in the MPEG-7 and MPEG-21 international standardization efforts, contributing to the definition of the MPEG-7 texture descriptor, the MPEG-21 DIA visual impairment descriptors, and modality conversion. His research interests include image/video processing, multimedia adaptation, visual data mining, image/video indexing, and multimedia security. Dr. Ro received the Young Investigator Finalist Award of ISMRM in 1992 and the Scientist Award in Korea in 2003. He served as a TPC member of international conferences such as IWDB, WIAMIS, AIRS, and CCNC, and he was the co-program chair of IWDD 2004.”