AUTOMATIC CHARACTER IDENTIFICATION IN FEATURE-LENGTH FILMS

Yi-Fan Zhang[1,2], *Changsheng Xu*[2], *Hanqing Lu*[1]

[1] National Lab of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
{yfzhang, luhq}@nlpr.ia.ac.cn
[2] Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613
xucs@i2r.a-star.edu.sg

ABSTRACT

This paper presents a novel approach to automatically identify characters in films using audio visual cues and text analysis. The approach consists of three stages: (i) frontal face track detection and clustering, (ii) face track classification, (iii) name assignment. A Finite State Machine (FSM) method is utilized to filter faces detected on each frame and build face tracks. The face tracks are clustered using constrained K-Centers. The tracks located in the center area of each cluster are set as exemplars. The marginal points of each cluster and the newly detected non-frontal face tracks are classified to these exemplars using complementary cues of audio and visual. The names of characters are ranked based on their occurrences in the film script and the face track clusters are ranked based on track counts. The names are assigned to the clusters according to the ranking order. Experiments were conducted on two feature-length films and gave promising results.

Index Terms— movie analysis, face recognition, speaker identification

1. INTRODUCTION

Identifying characters in films, although very intuitive to humans, still poses a significant challenge to computer methods. This is due to that characters may show variation of their appearances including scale, pose, illumination, expression and wearing in a film. Matching people based on their faces is a well known difficult problem even under quite controlled conditions; meanwhile, giving identity to recognized faces also needs to tackle the ambiguity of identities. The objective of this work is to identify characters present in films and label them with their names. This can be applied in many areas, such as character based video retrieval, personalized video summarization, intelligent playback, and video semantic mining, etc.

In a film, as characters carry a great much semantic information, they are always the major focus of interest to the audience. A lot of research work has been concentrated on character based film analysis. Major cast detection and automatic cast listing are main tasks among the existing work. Scattered faces of the same character were clustered together [1]. The major casts were determined by the face co-occurrence information [2] or accumulative temporal and spatial presence [3]. In feature-length films, since face recognition is difficult due to noisy environment, approaches only based on face are not always reliable. Therefore, multi-modal approaches using audio and visual cues were proposed [3][4]. However, these approaches cannot automatically assign real identities to each character. To handle this, textual information (subtitles and film scripts) was used to match the names to the faces in [5], but this approach only detected and labeled frontal faces. The subtitle text recognition using OCR and the alignment of subtitle and script may also introduce noises.

Fig. 1. Example face detections in the film "Notting hill". All the detected faces are normalized into 150×120 pixels images.

We present a novel approach to identify characters in feature-length films using audio visual cues and text analysis. Face and speaker voice features are employed for character classification. The names of characters are obtained from the readily available text: film script. The approach is consisted of three stages: (i) frontal face track detection and clustering, (ii) face classification, (iii) name assignment. Firstly, frontal faces in each frame are detected from the entire film. A FSM method is utilized to build the face tracks which are the basic granularity of the work. The tracks are clustered by constrained k-Centers within each episode of the film. The points located in the center area of each cluster are set as exemplars and used to segment audio tracks to train speaker voice models. The marginal points of each clusters and the newly detected non-frontal face tracks are classified into these clusters using face and speaker voice features. Then we rank these clusters based on the face track counts. In the script of the episode, the characters' names are also ranked based on their occurrences. Finally, the names are assigned to each cluster according to the ranking order. The context information is also used to tackle the ranking ambiguity. Comparing with the previous work, the contributions we bring here are: (i) extend the identification ability to non-frontal faces and (ii) propose a multi-modal framework integrating visual, audio and text for character identification.

2. FACE TRACK DETECTION AND CLUSTERING

We have two tasks in this stage: (1) detecting faces on each video frame and build face tracks from the detected faces, and (2) clustering the scattered face tracks which belong to the same character into the same group.

2.1. Detection

We detect frontal faces on each frame of the film using "boosted cascade" face detector [6], which shows certain robustness in the vari-

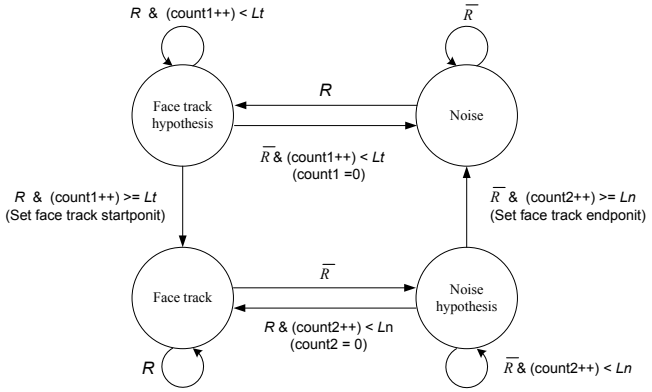

Fig. 2. Finite state machine for face track detection.

ation of scale, facial expression and illumination (see Figure 1). The frame number, position and scale of each detected face are stored. The accuracy of face detection is around 60%. Based on the coarse frame-level detection result, we employ a FSM to filter false positives and connect consecutive faces of one character to a face track. One face track is a continuous appearance of the faces of a character in the film. A typical film may contain up to 100000 faces detected on all the frames which are derived from a few thousands of tracks. Using face track as the granularity of our work can reduce the volume of data to be processed and preserve multiple exemplars for a character.

Face tracks are obtained using the FSM method. The sequence data of the position and scale of faces in each frame are used as the input and will pass a four state transition diagram as shown in Figure 2. The transition conditions between two states are labeled on each edge, and the corresponding actions are described in parentheses. In the figure, R means the faces on consecutive frames satisfy a rule defined as follows:

$$\begin{cases} \sqrt{d^2x + d^2y} < kr_0 \\ |\dfrac{dr}{r_0}| < k \end{cases} \tag{1}$$

where dx and dy are the differences of the center point's coordinates between current and previous face, and dr is the difference of the two radiuses. Here the detected face is supposed to be a circle region. r_0 is the previous face's radius. k is the parameter to control the detection confidence. $\bar{R}$ means the rule is not satisfied. $count_1$ and $count_2$ are the frame counts to measure the length of face track and noise respectively. L_t and L_n indicate the minimum length of face track and noise.

The faces which are continuous in position and scale are selected to build face tracks. Short tracks which are mostly created by the false positive face detections are filtered. Since short interruption in a track by the noise is allowed, the scheme is also robust to short-time occlusion and face pose changing.

2.2. Clustering

Although face can be considered as something unique and in some sense constant to a character, the appearance of it may be partially changed within the whole film by the character changing hair style or wearing glasses and hat. Thanks to the chapter information in DVDs, the whole film can be divided into a series of episodes. In an episode, it can be assumed that a character's appearance will not change. Hence, the face track clustering is conducted within each episode.

2.2.1. Similarity measurement

To measure the similarity of two faces, each face is represented as a quantitative feature vector. We extract SIFT descriptors on a face covering forehead, two eyes, nose and mouth and rejecting the points too close to the face region boundary. Each SIFT descriptor is an 8 bins histogram of image gradient orientations at a spatial 4×4 grid. Thus, it gives a 128 dimensional descriptor for each local feature position. About 30-50 descriptors are extracted for each face. We match them between two faces and set the number of matching descriptor pairs as their similarity $s(f_i, f_j)$. Since a face track is a set containing a sequence of faces, we define the similarity of two face tracks m and n by the maximum element-pair similarity between the two sets:

$$S(T_m, T_n) = \mu \cdot max_{i,j}(s(f_{m,i}, f_{n,j})) \tag{2}$$

where $f_{m,i}$ is the ith face in track m, $f_{n,j}$ is the jth face in track n, μ is used for normalization.

2.2.2. Constrained K-Centers clustering

A constrained K-Centers clustering is performed to group the scattered face tracks which belong to the same character. The temporal overlapping is implemented as a "cannot link" constraint while clustering: the two face tracks which have the identical frames cannot be grouped together. The number of clusters is set by prior knowledge derived from the script of the episode. We count the number of different speaker names occurring within this episode and set it as the number of clusters. This is based on our second assumption: the speech accompanies the appearance of the character. Here the voice-over in the films is not considered because they are labeled as "V. O." in the scripts and can be excluded by preprocess. The character without even one spoken line is also ignored and will not create a separate cluster. Each face track is assigned to the nearest cluster, including the unavoidable noises during the face track detection. Note that we will do cluster pruning in the next stage.

3. FACE TRACK CLASSIFICATION

The clustering in the first stage builds an initial discriminant space. Each cluster's center and its neighbors provide multiple exemplars that can generate discriminants and allow to project other points into this space. Hence, in this stage, our task is to classify those marginal points of each cluster and the newly detected non-frontal face tracks to each cluster depending on comparison with those exemplars. The multi-modal features are fused (i.e. the face and the speaker voice) to enhance the classification.

3.1. Cluster pruning

Based on the clustering result, we wish to clean those false alarms in face track detection and the marginal points which have low confidence to have the same identity with the cluster center. Firstly, we remove small clusters which contain fewer than 5 face tracks, as they are probably clustered by the noise. Then, within each cluster, we remove marginal points and reserve the left as exemplars of this cluster for later classification. The marginal points are determined

by assessing how confident they belong to their cluster:

$$Conf(T) = \frac{K_{in}}{K} S(T, T_c) \tag{3}$$

where $S(T, T_c)$ is the similarity between the face track T and its cluster's center T_c. K_{in} is the number of T's nearest neighbors which belong to the same cluster with T, K is the total number of T's nearest neighbors. The point whose confident value is lower than a threshold will be regarded as the marginal one and removed.

3.2. Voice model building

After pruning, there left the face tracks with high confidence belonging to the same character. They are used to set multiple face exemplars and meanwhile to segment audio tracks to build speaker voice models for each cluster. As the face track's start and end frame number are stored during face track detection, we can collect the corresponding audio segments. First, these audio segments need to be classified into speech and non-speech. For speech/non-speech classification, each segment is divided into a frame sequence, in which each frame is 50ms long and overlaps with the previous frame by 25ms. Five types of frame-level audio features are extracted: MFCC, LPC, LPCC, Zero Crossing Rate, and Short Time Energy. We combine them into a feature vector and employ Adaboost method to select the most discriminating dimensions of the vector to train classifier [7]. The class of each segment is determined by a majority voting of its frames. After classification, it still has ambiguities because there might be temporal overlapping between face tracks (i.e. different faces present in identical frames) and thus they may share or partially share one speech segment. Hence, we need to determine the dominant face track which has a higher correlation to this speech segment. Based on observation, we found that those face tracks which have larger scale's face or less overlapping with other tracks are more likely to be the real speaker. A dominant score of a face track can be defined as follows:

$$D = \frac{L}{L_{track}} Sigmoid(\bar{r}) \tag{4}$$

where L_{track} is the length of the face track, L is the length of the track not overlapping with others, $\bar{r}$ is the average radius of the faces in the track. In a cluster the speech segments of those face tracks whose dominant score exceed a threshold are collected as training set to train a speaker voice model.

Gaussian mixture models (GMMs) are employed here as it has been proved successful in speaker recognition application. A Gaussian mixture density is a weighted sum of M component densities given by:

$$p(\vec{x}|\lambda) = \sum_{i=1}^{M} w_i p_i(\vec{x}) \tag{5}$$

where $p_i(\vec{x})$ is the ith unimodal Gaussian densities, $\vec{x}$ is a 39-dimensional feature vector (i.e. 13-dimensional MFCCs and their temporal delta, accelerate values for each frame of the speech segment), the mixture weights further satisfy the constraint $\sum_{i=1}^{M} w_i = 1$. Under the assumption of independent feature vectors, the log-likelihood of a model λ for a sequence of feature vectors $X = \{\vec{x_1}, ..., \vec{x_T}\}$ is computed as follows:

$$\log p(X|\lambda) = \frac{1}{T} \sum_{t} \log p(\vec{x_t}|\lambda) \tag{6}$$

where $p(\vec{x_t}|\lambda)$ is computed as in Equation 5. The speaker whose model gives the maximum likelihood is determined as the target speaker.

3.3. Classification to exemplars

We collect those marginal points in each cluster and the newly detected non-frontal face tracks from the film for classification. The non-frontal face tracks are detected using the same way as the frontal face tracks. The only difference is that the faces on each frame are detected by the "boosted cascade" detector trained by non-frontal faces. For classification, as we have learnt discriminate functions from face and voice features, we adopt late fusion [8] to combine these scores and yield a final classification score. Let C_k be the kth cluster which reserves N face track exemplars and one speech voice model λ_k. Let T be the face track to be classified, X be T's corresponding audio segment's feature vector. The final discriminate function is defined as follows:

$$\begin{cases} F(T, C_k) = \alpha Max_n(S(T, T_{k,n})) + \beta D_T \log p(X|\lambda_k) \\ \alpha = 0.5, \beta = 0.5, \qquad \text{(if X is speech)} \\ \alpha = 1.0, \beta = 0.0, \qquad \text{(if X is not speech)} \end{cases} \tag{7}$$

where $S(T, T_{k,n})$ is the similarity between T and the nth exemplar track of C_k, D_T is the T's dominant score, $\log p(X|\lambda_k)$ is the likelihood of C_k's voice model λ_k for X. Note that if X is determined as non-speech, β will be set to 0. The face track will be classified into the cluster whose function score $F(T, C_k)$ is maximal.

4. NAME ASSIGNMENT

After clustering and classification, we obtain clusters in which all the face tracks belong to the same character. To recognize the character corresponding to each cluster, we need to assign names to them. The names are obtained from the film script by name entity recognition. In the script, there are speakers' names before their spoken lines. We rank the occurrences of different names in the script and the size of each cluster (i.e. the face track's count), and assign the names to clusters according to the ranking order. Let m be the number of clusters and n be the number of names, where $m \leq n$, as we have removed some too small clusters. Hence, we assign the top m ranked names to the m clusters. Although the occurrence counts of a character's names in the script and his/her face tracks in the film are not exactly identical, it basically follows that the more spoken lines, the more occurrences of the character in the film. Take the episode "birthday party" in the film "Notting hill" for instance, Figure 3 shows the ranking of occurrence counts of the face tracks in the film and the names in the script.

In some circumstances, some cluster's sizes are very similar or even the same. Hence, those clusters whose sizes' difference is less than 5 counts (e.g. the clusters of "Bella" and "Bernie" showing in Figure 3) are set as pending status. We refuse to determine their

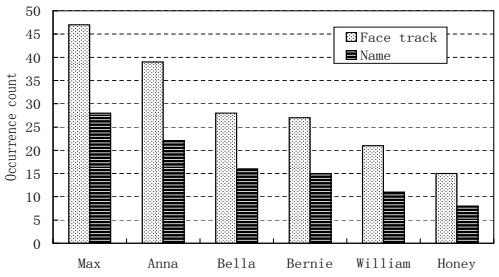

Fig. 3. Rank of face tracks and names in one episode.

names immediately because their ranking order may be affected by misclassification. Their names are determined later by their contextual face tracks in the film which have been recognized. In the pending cluster, we collect each track's contextual tracks, which are temporal neighboring or overlapping with it. Then we find the most frequent identity $name_i$ among the contextual tracks. Meanwhile, in the script, the contextual names of each name are also recorded respectively. Finally, the name, whose most frequent contextual name is $name_i$, is detected and used to label the pending cluster.

5. EXPERIMENTS

The proposed approach was applied on two feature-length films: "Notting hill" and "The Shawshank redemption". In "Notting hill", 105792 frontal faces and 25621 non-frontal faces were detected respectively on the total 178446 frames of the film. In "The Shawshank redemption", 82835 frontal faces and 18657 non-frontal faces were detected on 189090 frames. To evaluate the FSM method for face track detection, experiments are conducted on totally 90 minutes long clips segmented from the two films. The confidence parameter k in Equation 1 is tuned to get detection results at different levels of recall. Since it is difficult to definitely distinguish the frontal and non-frontal faces while manually labeling ground truth, we mixture them together and compute the whole detection precision and recall which are shown in Figure 4. When k is 0.10, the recall is 0.88 and the precision is 0.95. We can see that more noises will be included if the recall achieves higher. Hence, we set k = 0.10 and used it on the two entire films. In "Notting hill", 1466 frontal and 337 non-frontal face tracks were built from the detected faces. In "The Shawshank redemption", 1157 frontal and 292 non-frontal face tracks were built.

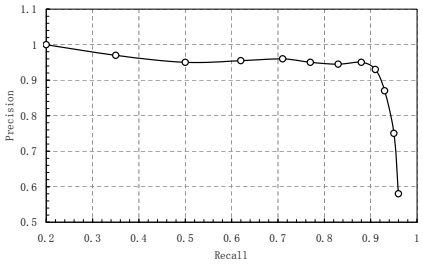

Fig. 4. Presicion/recall curve for face track detection.

To demonstrate the performance of the character identification approach, the precision/recall curves are shown in Figure 5. The recall means the proportion of tracks assigned with a name against the whole tracks occurring in the film. The precision means the proportion of tracks with a correct name against the whole tracks with a name. We give a threshold T to the classification function score in Equation 7. If the scores of a face track with all the clusters are lower than T, this track will not be classified into these clusters and thus not be labeled with a name. By changing T from 0.4 to 0.1, we can obtain identification precisions at different levels of recall. To compare with the proposed approach, we remove the audio features and use only faces in the face track classification stage. The results are also shown in Figure 5. As expected, it can be seen that using complementary cues of face and audio, we can achieve higher identification precision than using only faces at the same levels of recall.

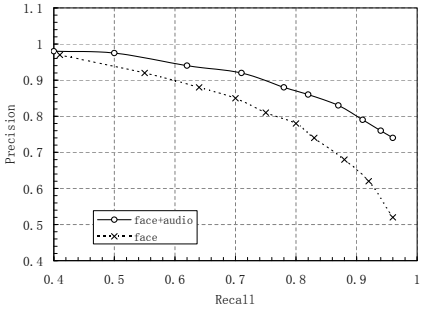

Fig. 5. Precision/reacll curves for character identification.

6. CONCLUSION

In this paper, we have proposed an approach for character identification in feature-length films. Multi-modal information of visual, audio and text are integrated for character classification and name assignment. The identification ability has been extended to non-frontal faces and achieve promising result. Based on current work, we intend to develop more sophisticate metric to measure face similarity and incorporate face pose estimation module to enhance the face classification result.

7. ACKNOWLEDGEMENT

The work is supported by the 863 Program of China (Grant No. 2006AA01Z315, 2006AA01Z117).

8. REFERENCES

[1] A. Fitzgibbon and A. Zisserman, "On affine invariant clustering and automatic cast listing in movies," in *Proceedings of ICCV*, 2002, vol. 3, pp. 304–320.

[2] Y. Gao et al, "Cast indexing for videos by ncuts and page ranking," in *Proceedings of CIVR*, 2007, pp. 441 – 447.

[3] Z. Liu and Y. Wang, "Major cast detection in video using both speaker and face information," *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 89–101, 2007.

[4] Y. Li, S. Narayanan, and C.-C. Jay Kuo, "Content-based movie analysis and indexing based on audiovisual cues," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 8, pp. 1073– 1085, 2004.

[5] Mark Everingham, Josef Sivic, and Andrew Zisserman, ""hello! my name is... buffy" automatic naming of characters in tv video," in *Proceedings of BMVC*, 2006.

[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of Computer Vision and Pattern Recogntion*, 2001, pp. 511–518.

[7] Y.F. Zhang, Q.S. Liu, J. Cheng, and H.Q. Lu, "Multimodal based highlight detection in broadcast soccer video," in *Proceedings of Asia-Pacific Workshop on Visual Information Processing*, 2007.

[8] C. Snoek, M. Worring, and A. Smeulders, "Early versus late fusion in semantic video analysis," in *Proceedings of ACM on Multimedia*, 2005, pp. 399–402.