Character Relationship Analysis in Movies Using Face Tracks

Yi-I Chiu, Pau-Choo Chung
National Cheng Kung University
Tainan, Taiwan
chillychiu@gmail.com, pcchung@ee.ncku.edu.tw

Chun-Rong Huang
National Chung Hsing University
Taichung, Taiwan
crhuang@cs.nchu.edu.tw

Abstract

In this paper, we propose using face tracks to model faces of the same character under variant head motions and scene illuminations. A novel measurement is developed to assess the similarity between two face tracks with different lengths. Then, based on temporal constraints, the character relationship graph of a movie is built. As shown in the experiments, our method can successfully retrieve character relationships from movies in real-time without any prior information or training.

1 Introduction

Exploring character relationships from videos provides a cognitive way for content understanding and indexing. To achieve the goal, state-of-the-art methods [1][2][3] usually analyze temporal structures of videos, i.e. shots and scenes [4], at first. In general, a shot indicates continuous presence of characters and a scene contains interactions of characters in a relatively larger social group. Therefore, extracting shots and scenes are considered as effective preprocessing for analyzing character relationships in movies. To identify the same characters in different shots, face detection is performed for each shot. Then, face recognition based [1][3] and face clustering based [2] methods are employed to identify repeatedly appearing characters. Finally, character relationships are obtained by using temporal scene constraints. Detailed reviews of character relationship analysis methods can be found in [5].

Besides using face information for character relationship analysis, some researchers combine transcripts [6], and cast lists [7] with face information to identify people. As mentioned above, most of the state-of-the-art methods retrieve character relationships based on temporal video segmentation and face recognition results. However, current scene detection methods such as [8] hardly perform promising results on different styles of movies. Moreover, because of variant head motions and scene illumination conditions in shots, the recognized characters are not accurate enough for character relationship analysis.

In practice, when watching movies, the audience can understand character relationships without knowing any temporal video structures. The interactions among characters (appear consecutively or in the same frame) provide audience sufficient evidence to interpret character relationships in the movie. Thus, we propose a novel bottom-up interpretation of videos as follows: face, face tracks, and face groups to automatically construct character relationship graphs. Here a face track is defined as a collection of consecutive faces obtained by using face tracking algorithm with position and appearance priors, which will be used to represent each person. A face group is the union of face tracks of the same character appearing in the movie.

To obtain the representation, the video is firstly processed by face detection and face tracking algorithms to obtain face tracks appearing in the video. Contrast-based face features are employed to represent these unaligned faces. One of the major problems to define the similarity between two face tracks is that the lengths of two face tracks are usually different. Although closest face pairs [6] and earth mover's distance (EMD) [2] of face track distributions were proposed to solve the problem, these methods are still hard to model the real distributions of face tracks and lead false matching results. We propose a new face track similarity assessment based on the concept of principle angles [9] to solve the length problem. With the face track similarity, agglomerative clustering algorithms [10] is then applied to merge face tracks into face groups. Finally, the character relationship graph is formed by linking consecutively appearing face groups.

2 Face and Face Tracks

A face track is defined as a group of spatially and temporally consecutive faces which contains variant head motions and illumination changes. Most methods align faces to its frontal view for matching. However, face alignment is time consuming. Moreover, non-frontal faces can also provide face descriptions for identifying characters.

To obtain face tracks, face detection [11] is performed to find the first appearing face from the video. Once a face is detected in the t-th frame, we start the face tracking procedure using the position prior to reduce face detection time. The position prior defines ROI in frame $t+1$ of the detected face in frame t. If a face f_t that appears in frame t overlays a face f_{t+1} in frame $t+1$, f_{t+1} is considered as a candidate tracked face of f_t. The appearance prior is proposed to verify if the tracked face belongs to the same person. If a candidate face f_{t+1} is visually similar to f_t, it is then confirmed to be the following face of f_t.

In general, appearances of faces in different shots of a person vary a lot with different orientations, lightings, and facial expressions. Fortunately, a video contains continuous motions of people. The appearance changes of the faces of a person between adjacent frames will be relatively minor. Here, we use modified contrast context histogram (CCH) [12] descriptors to efficiently represent the appearance prior of face regions.

For each detected face f_t, an oval mask is firstly applied to the face region. The masked face region is normalized to zero-mean and unit variance. Seven reference keypoints $[K_1, \ldots, K_p, \ldots, K_7]$ and the surrounding region q_{K_p} for each K_p are used to represent

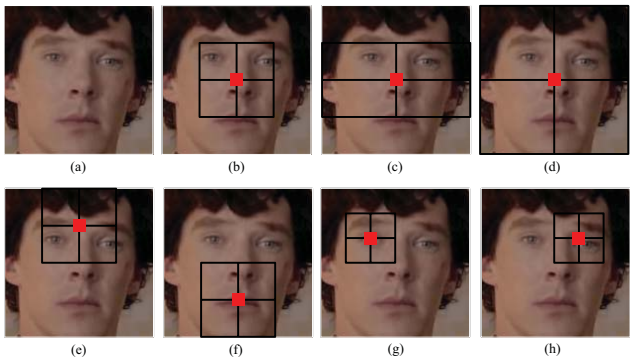


Figure 1. Seven sampled keypoints (red dots) and their surrounding face regions (black rectangles). (a) the original face image, (b)-(h) keypoint K_1-K_7.

seven dominant face regions as shown in Fig. 1. The keypoint locations in Fig. 1(b)-(d) are located at the center of the face region with different sizes of surrounding regions. K_4-K_7, as shown in Fig. 1(e)-(h), represent the upper part, lower part, upper left and upper right of the face. For each keypoint K_p, CCH descriptors in RGB channels are generated as follows.

The surrounding region $q^C_{K_p}$ of K_p in channel C is divided into 2×2 grids, upper-left ul, upper-right ur, lower-left ll, and lower-right lr. The average of nine pixels around the keypoint K_p in each channel is referred as $r^C_{K_p}$. For each grid in each color channel, two contrast features are computed with respect to $r^C_{K_p}$ as follows:

$$F(q^C_{K_p,ul}, +) = \frac{1}{q^C_{K_p}} \sum_{p \in q_{K_p,ul}} \mathbf{1}_{\{p|p \geq r^C_{K_p}\}},$$

and

$$F(q^C_{K_p,ul}, -) = \frac{1}{q^C_{K_p}} \sum_{p \in q_{K_p,ul}} \mathbf{1}_{\{p|p < r^C_{K_p}\}}.$$

Total $7 \times 3 \times 2$ features are generated for each face image. These features are denoted as $F(f)$ and normalized to $\|F(f)\| = 1$ to achieve linear illumination invariance. With the face appearance features $F(f_t)$ and $F(f_{t+1})$, we define the distance between two faces f_t and f_{t+1} as follows:

$$D(f_t, f_{t+1}) = \arccos(F(f_t)^T F(f_{t+1})).$$

If the distance $D(f_t, f_{t+1})$ exceeds the threshold th, the track is split because f_t and f_{t+1} might belong to different characters. In order to avoid mis-tracking of newly appearing faces in a frame (ex. a group of people are talking and someone walks in), face detection is performed on the whole image every 10 frames. If newly incoming faces are detected, a new tracking process will start to track them. Each face track T_i is expressed as $[F_{i,a_i}, \ldots, F_{i,t}, \ldots, F_{i,b_i}]$, where a_i and b_i are the starting and ending frame indices of T_i, respectively. Given a video, we obtain M face tracks $[T_1, \ldots, T_i, \ldots, T_M]$.

3 Face Group Generation

By considering position and appearance priors, each face track represents continuously moving faces of a character in the movie. Automatically identifying face tracks of a character is an important issue for obtaining correct character relationship graph. Face recognition is one of the most popular approaches to handle this issue. Each face in the face track is recognized based on classifiers trained from labeled face data. To obtain better face recognition results, face alignment is required. Then, given an appearing face, it is aligned with face training data to obtain recognition results. Three issues need to be clarified. Firstly, the collected manually labeled face training data are hard to completely model the faces under different illuminations and head motions in the video. Secondly, a frontal aligned face is necessary to achieve better recognition results. Thirdly, most face recognition methods consider recognizing one face image at a time which do not use the temporal face information. Due to variations of different videos, it is deficient to apply face recognition to identify repeatedly appearing people.

3.1 Face Track Similarity

Because above mentioned issues, identifying the same character using one face image by face recognition methods is unreliable [1]. Unlike a face image, a face track contains continuous face images under different head motions and scene illuminations. Thus, it can represent both frontal faces and profiles of the character at the same time. Moreover, when the scene illumination changes gradually, each face in the face track will also record the illumination changes. Thus, it can be considered as containing sampled faces from the face distribution under variant illumination changes. As a result, face tracks can provide much more evidence for character identification in movies.

Comparing similarity between two face images is easy, because their feature dimensions are the same. However, measuring the similarity between two face tracks with different lengths is not straightforward. Recently, canonical correlations (CC) [9] is proposed to describe the similarity between two unordered image sets. Specifically, canonical correlations measure the consistence of two linear subspaces by the smallest principal angles between the subspaces. Such concept inspires us to define the similarity between two face tracks by using principal angles.

Given two face tracks T_i and T_j, we aim to define the distance $D(T_i, T_j)$ between T_i and T_j. Please note that the lengths N_i and N_j of T_i and T_j are different. To define the face track similarity, we consider a relaxed constraint: if two face tracks have sufficient number of similar faces, they are more likely to belong to the same character. Principal angles are used to evaluate the number of similar faces between two face tracks. The principal angles of T_i with respect to T_j is defined as follows:

$$\theta_k(T_i, T_j) = \min_{k'} \{\arccos(F^T_{i,k} F_{j,k'}) \mid F_{i,k} \in T_i, F_{j,k'} \in T_j\},$$

where the number of principal angles is N_i. Because the lengths of two face tracks are different, $\theta_k(T_i, T_j)$ does not necessary equal to $\theta_k(T_j, T_i)$.

According to principal angles, the number of similar faces of T_i with respect to T_j is defined as follows:

$$\|\theta_k(T_i, T_j)\|_0 = \#\{k|\theta_k(T_i, T_j) \leq d^*\},$$

where d^* is the similarity upperbound that ensure two faces belong to the same character,

$$d^* = \max\{d \mid \forall D(F, F') < d, F \in P, F' \in P\}.$$

We then utilize the cumulative similarity between two face tracks to determine whether these tracks belong to the same character. To measure the track consistency of face track T_i, we define a threshold for track consistency n_i based on the length of the face track and characteristic of the descriptor. As a result, the distance between two face tracks T_i and T_j is defined as follows:

$$D(T_i, T_j) = \min\{d^* \mid \|\theta_k(T_i, T_j)\|_0 > n_i, \\ \|\theta_l(T_j, T_i)\|_0 > n_j\}$$

If faces are aligned, we can choose smaller n_i. If faces are not aligned, we require more information to determine if two face tracks are alike, therefore larger n_i is chosen. Shorter face track will have smaller n_i. Even the lengths of face tracks vary greatly, the face track similarity can still be uniquely and efficiently computed. With the face track similarity, agglomerative clustering algorithms [10] is then applied to merge face tracks into face groups.

4 Character Relationship Graph Construction

After obtaining face groups of characters in the movie, interactions of characters are used to explore their social relationships. Two face group G_u and G_v are considered to have certain social relationships if they interact with each other in the video. In general, the interactions of character in the movie can be categorized as follows. The first type of interactions, R_o, is that two or more people co-occur in the same frames, i.e. their face tracks overlay in the temporal domain. The second type of interactions, R_c, is that two or more people have a conversion. The face tracks of these people will alternatively switch in certain temporal orders. For example, character P_1 appears at first and then character P_2 appears. These two kinds of interactions provide strong social relationship connections. An interesting case is that P_1 is followed by P_2 and this situation only occurs once, denoted by R_n. Since P_1 and P_2 are temporally connected in the video, they may have social relationships. However, compare to the conversion case, interactions between P_1 and P_2 in this case are not guaranteed. Thus, we consider such neighbor face track cases as weak social relationships which imply that P_1 and P_2 might correlate to each other. Finally, the character relationship graph is generated by linking face groups via R_o, R_c and R_n.

5 Experimental Results

To evaluate the proposed method, two commercial videos are used. For comparison, we implement the features proposed in [6] to construct the character relationship graph for each video. Please note that face alignment is performed in [6] so that the obtained face keypoints are correctly located on the frontal faces of each face track. In contrast, no face alignment is performed to obtain face tracks in our method. Fig. 2(a) shows the ground truth of the character relationship of the first video. People who have social relationships are connected by black lines. The width of the black line indicates the interaction degree between two people. The bolder black line implies that two people interact with each other more often. Fig. 2(b) and (c) show the generated social relationship graphs of using features in [6] and the proposed method. The red, green, and blue links represent the social connections of R_c, R_o, and R_n, respectively. If a face group contains misclustering results, two or more face images of different people will be in the same rectangle.

As shown in Fig. 2(b), when using the features in [6], face tracks of different people under the same orientations are merged to the same face group. This situation implies that face orientations will affect the similarity between face tracks. Compared to the ground truth, although the number of finally obtained face groups is larger than the number of people, the social connections are still successfully linked by considering co-occurring, conversion and neighborhood constraints in the proposed method. Fig. 3 also shows similar results as Fig. 2. Even a person has multiple face groups, most social connections still correctly link to correlated face groups of another people for both methods. Our method is implemented using Matlab on an Intel i7 computer with a 3.4G CPU and 4G memory. The averaged computation time per frame of test videos for each step of our method and features in [6] is 54.9 and 181.4 milliseconds.

6 Conclusion

Representing characters in the movies via face tracks can avoid the temporal video segmentation problem. Moreover, face tracks also provide better face representation for character identification and character relationship graph construction.

7 Acknowledgment

This work was supported in part by the National Science Council of Taiwan, R.O.C., under NSC-101-2221-E-005-086-MY3 and NSC-101-2221-E-006-262-MY3.

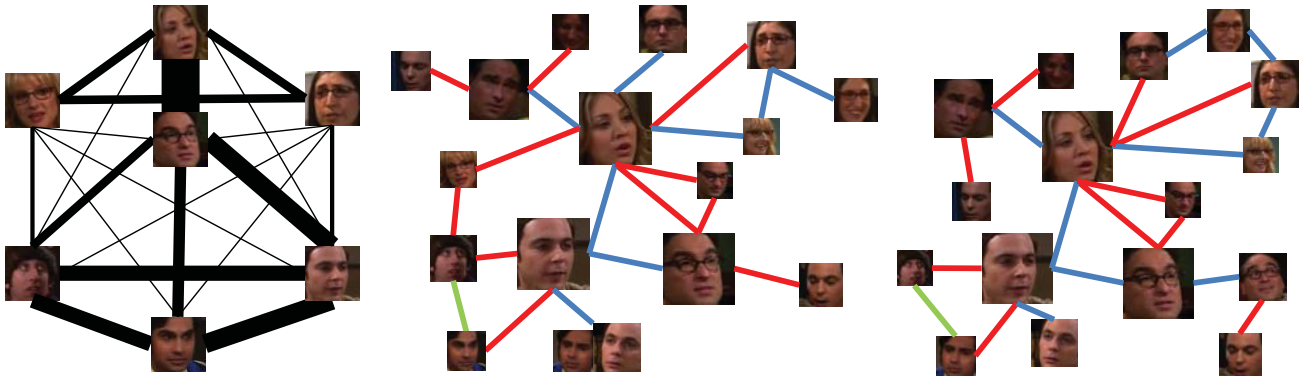
References

[1] C.-Y. Weng, W.-T. Chu, and J.-L. Wu, "Rolenet: Movie analysis from the perspective of social networks," *IEEE Trans. on Multimedia*, vol. 11, no. 2, February 2009.

[2] Y.-F. Zhang, C. Xu, H. Lu, and Y.-M. Huang, "Character identification in feature-length films using global face-name matching," *IEEE Trans. on Multimedia*, 2009.

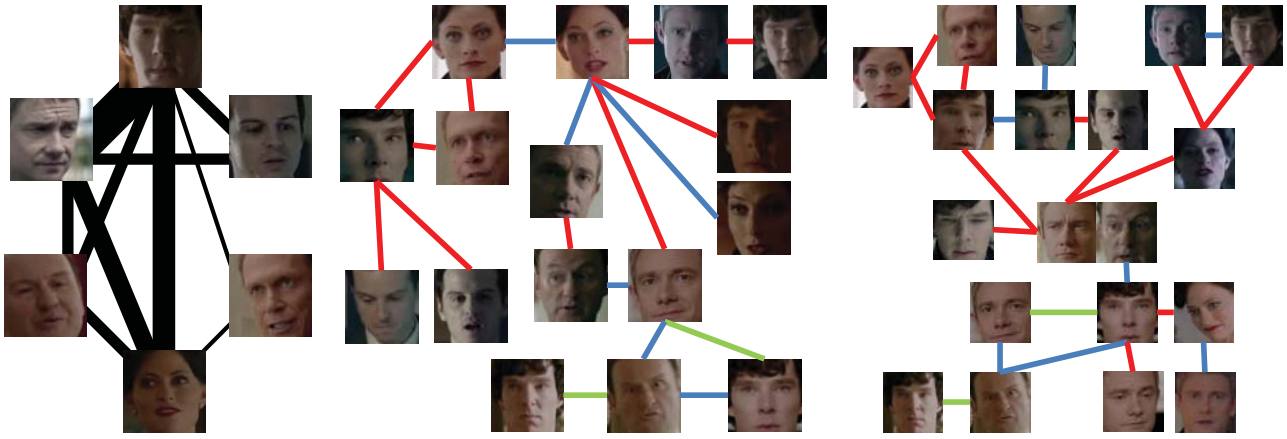
[3] L. Xie, A. Natsev, M. Hill, J. R. Kender, and J. R. Smith, "Visual memes in social media: Tracking real-world news in youtube videos," in *In Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 53–62.

[4] Z. Xiong, X. Zhou, Q. Tian, R. Yong, and T. S. Huang, "Semantic retrieval of video - review of research on video retrieval in meetings, movies and broadcast news,



(a) (b) (c)

Figure 2. The social relationship graphs of video 1 generated by (a) ground truth. (b) features in [6]. (c) the proposed method. The yellow, green, and blue links represent the connections of R_c, R_o, and R_n, respectively.



(a) (b) (c)

Figure 3. The social relationship graphs of video 2 generated by (a) ground truth. (b) features in [6]. (c) the proposed method. The yellow, green, and blue links represent the connections of R_c, R_o, and R_n, respectively.

and sports," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 18–27, 2006.

[5] H. Sundaram, L. Xie, M. D. Choudhury, Y.-R. Lin, and A. Natsev, "Multimedia semantics: Interactions between content and community," in *Proceeding of the IEEE*, vol. 100, no. 9, Sept. 2012, pp. 2737–2758.

[6] 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*, 2006.

[7] M. Xu, X. Yuan, J. Shen, and S. Yan, "Cast2face: Character identification in movie with actor-character correspondence," in *in Proceedings of ACM Multimedia*, 2010, pp. 831–834.

[8] Z. Rasheed and M. Shah, "Detection and representation of scenes in videos," *IEEE Trans. on Multimedia*, vol. 7, no. 6, pp. 1097–1105, 2005.

[9] A. Bjorck and G. H. Golub, *Numerical methods for computing angles between linear subspaces.* Mathematics of Computation, 1973, vol. 27, no. 579-594.

[10] B. S. Everitt, S. Landau, and M. Leese, *Cluster Analysis.* London: Arnold., 2001.

[11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[12] C.-R. Huang, C.-S. Chen, and P.-C. Chung, "Contrast context histogram - an efficient discriminating local descriptor for object recognition and image matching," *Pattern Recognition*, vol. 41, no. 10, pp. 3071–3077, 2008.