Character Identification in Feature-Length Films Using Global Face-Name Matching

Yi-Fan Zhang, *Student Member, IEEE*, Changsheng Xu, *Senior Member, IEEE*, Hanqing Lu, *Senior Member, IEEE*, and Yeh-Min Huang, *Member, IEEE*

***Abstract*—Identification of characters in films, although very intuitive to humans, still poses a significant challenge to computer methods. In this paper, we investigate the problem of identifying characters in feature-length films using video and film script. Different from the state-of-the-art methods on naming faces in the videos, most of which used the local matching between a visible face and one of the names extracted from the temporally local video transcript, we attempt to do a global matching between names and clustered face tracks under the circumstances that there are not enough local name cues that can be found. The contributions of our work include: 1) A graph matching method is utilized to build face-name association between a face affinity network and a name affinity network which are, respectively, derived from their own domains (video and script). 2) An effective measure of face track distance is presented for face track clustering. 3) As an application, the relationship between characters is mined using social network analysis. The proposed framework is able to create a new experience on character-centered film browsing. Experiments are conducted on ten feature-length films and give encouraging results.**

***Index Terms*—Face identification, movie analysis, social network analysis, video browsing.**

I. Introduction

With the flourishing development of the movie industry, a huge amount of movie data is being generated everyday. It becomes very important for a media creator or distributor to provide better media content description, indexing and organization, so that users can easily browsing, skimming and retrieving the content of interest. In a film, characters are the focus center of interests from the audience. Their occurrences provide meaningful presentation of the video content. Hence, characters are one of the most important content to be indexed, and thus character identification becomes a critical step on film semantic analysis.

Manuscript received August 04, 2008; revised April 12, 2009. First published August 18, 2009; current version published October 16, 2009. This work was supported in part by the National Natural Science Foundation of China (Grant No. 60833006) and in part by the Beijing Municipal Laboratory of Multimedia and Intelligent Software Technology. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jie Yang.

Y.-F. Zhang, C. Xu, and H. Lu are with the National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the China-Singapore Institute of Digital Media, Singapore, 119615 (e-mail: yfzhang@nlpr.ia.ac.cn; csxu@nlpr.ia.ac.cn; luhq@nlpr.ia.ac.cn).

Y.-M. Huang is with the National Cheng-Kung University, Tainan, Taiwan (e-mail: huang@mail.ncku.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2009.2030629

Character identification in feature-length films, although very intuitive to humans, still poses a significant challenge to computer methods. This is due to the fact that characters may show variation of their appearance including scale, pose, illumination, expression and wearing in a film. People recognition based on their faces is a well-known difficult problem [1]; meanwhile, giving identities to the recognized faces also needs to tackle the ambiguity of identities. The objective of this work is to identify the faces of characters in the film and label them with their names. Based on our work, users can easily use the name as a query to select the characters of interest and view the related video clips. This character-centered browsing is able to not only bring us a new viewing experience, but also provide an alternative for video summarization and digestion.

In this paper, we present a novel approach for character identification in feature-length films. In films, the names of characters seldom directly appear in the subtitle, while the film script which contains names does not have time stamps to align with the video. There are not enough temporally local name cues that can be found for local face-name matching. Hence, we attempt to do a global matching between the faces detected from the video and the names extracted from the film script, which is different from the state-of-the-art methods on naming faces in the videos. Based on the results of character identification, an application for character-centered film browsing is also presented which allows users to use the name as a query to search related video clips and digest the film content.

A. Related Work

The crux of the problem on associating faces with names is to exploit the relations between videos or images and the associated texts in order to label the faces with names under less or even no manual intervention. Extensive research efforts have been concentrated on this problem. Name-it [2] is the first proposal on face-name association in news videos based on the co-occurrence between the detected faces and names extracted from the video transcript. A face is labeled with the name which frequently co-occurs with it. Named Faces system [3] built a database of named faces by recognizing the people names overlaid on the video frames using video optical character recognition (VOCR). Yang *et al.* [4], [5] employed the closed caption and speech transcript, and built models for predicting the probability that a name in the text matches to a face on the video frame. They improved their methods in [6] by using multiple instance learning for partial labeled faces to reduce the effort of collecting data by users. In [7], the speech transcript was also used to find people frequently appearing in the news videos. Similarly, for face identification in news images, the problem

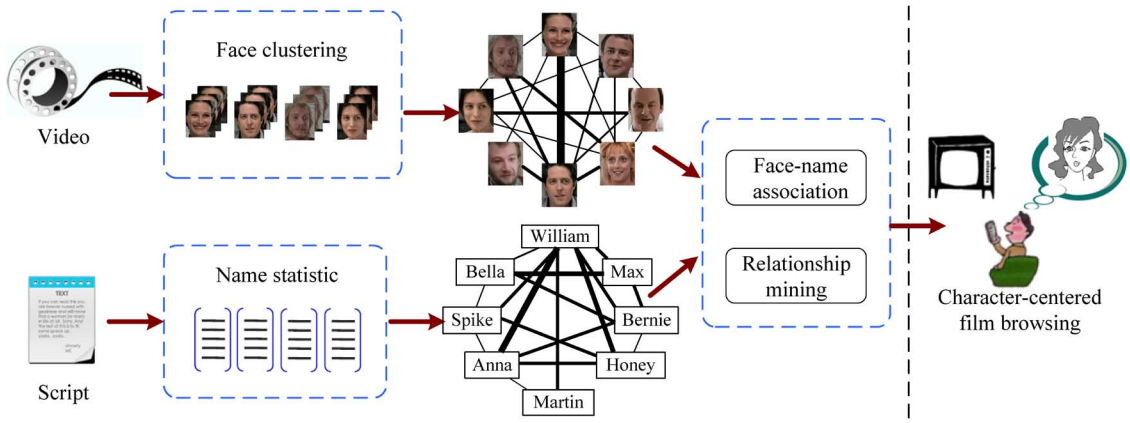


Fig. 1. Framework of character identification using global face-name matching.

was also addressed as clustering or classifying the faces to the people specific appearance models, supervised by the name cues extracted from the image captions or associated news articles [8]–[10].

Although some of the methods showed promising face identification results in the video, most of them are used in news videos which can easily get candidate names for the faces from the simultaneous appearing captions or temporally local transcripts. Unlike the news videos which are presented by the third person such as the anchor or the reporter, in the films, the names of characters are seldom directly appearing in the subtitle, which makes it difficult to get the local name cues. Hence, many efforts on film analysis were devoted to major characters detection or automatic cast listing but not assigning real names to them. Arandjelovic and Zisserman [11] used face image as a query to retrieve particular characters. Affine warping and illumination correcting were utilized to alleviate the effects of pose and illumination variations. In [12], multiple face exemplars were obtained from face tracks to improve the face matching results. For automatic cast listing, faces of the major characters in a feature film can be generated automatically using clustering based on the appropriate invariance in the facial features [13]–[15]. Due to the uncontrolled conditions in films with a wide variability on faces, approaches only depending on faces are not always reliable. Therefore, multi-modal approaches fused with facial features and speaker voice models were proposed [16], [17]. However, these approaches cannot automatically assign real names to the characters. To handle this, Everingham *et al.* [18] proposed to employ a readily available textual source, the film script, which contains the character names in front of their spoken lines. However, the film script does not have time information to achieve face name matching. Hence, they used the film script together with the subtitle for text video alignment and thus obtained certain annotated face exemplars. The rest of the faces were then classified into these exemplars for identification. Their approach was also followed by several works which aimed for video parsing [19] and human action annotation [20]. However, in their approach [18], the subtitle text and time-stamps were extracted by OCR, which required extra computation cost on spelling error correction and text verification. Sometimes, the cross-linguistic problem and the inconsistencies between subtitles and scripts may bring more difficulties in alignment.

B. Overview of Our Approach

In a film, the interactions among the characters resemble them into a relationship network, which makes a film be treated as a small society [21]. Every character has his/her social position and keeps a certain relationship with others. In the video, faces can stand for characters and the co-occurrence of the faces in a scene can represent an interaction between characters. Hence, the statistical properties of faces can preserve the mutual relationship in the character network. As the same way in the film script, the spoken lines of different characters appearing in the same scene also represents an interaction. Thus, the names in front of the spoken lines can also build a name affinity network. Both the statistical properties of the faces and the names motivate us to seek a correspondence between the face affinity network and the name affinity network. The name affinity network can be straightforwardly built from the script. For the face affinity network, we first detect face tracks in the video and cluster them into groups corresponding to the characters. During the clustering, the Earth Mover's Distance (EMD) is utilized to measure the face track distance. Since we try to keep as same as possible with the name statistics in the script, we select the speaking face tracks to build the face affinity network, which is based on the co-occurrence of the speaking face tracks. For name and face association, it is formulated as a problem of matching vertices between two graphs. A spectral method, which has been used in 2-D/3-D registration and object recognition, is introduced here to build name-face association. Especially, during the matching process, priors can be incorporated for improvement. After assigning names to faces, we also determine the leading characters and find cliques based on the affinity network using social network analysis [22]. A platform is presented for character-centered film browsing, which enables users to easily use the name as a query to search related video clips and digest the film content. The whole framework of our proposed approach is shown in Fig. 1.

Compared with the previous work, the contributions of our work include: 1) A graph matching method is introduced to

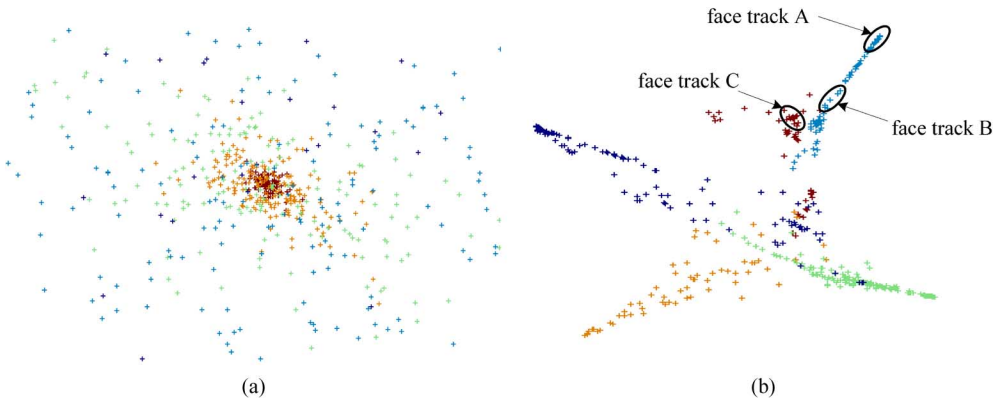


Fig. 2. Face images mapped into the embedding space described by the first two components of (a) PCA and (b) LLE. The same color points are the faces of the same character. In panel (b), the ellipses label three face tracks belonging to two characters.

build face-name association. 2) An EMD-based measure of face track distance is presented for face track clustering. 3) Based on character identification, the relationship between characters is mined and a platform is provided for character-centered film browsing. Although the graph matching method and the EMD measure are derived from the existing work, to the best of our knowledge, they have not been applied in the face naming problem and face track distance measurement yet. We choose them reasonably and integrate them in a novel solution to address the existing challenge problem: people identification in real-world videos. The global matching solution is different from the previous methods which are based on local matching.

II. Face Clustering

We first use a multi-view face tracker [23] to detect and track faces on each frame of the video. One feature-length film may contain up to 100 000 faces detected on all the frames, which are derived from a few thousands of tracks. Using face track as the granularity can reduce the volume of data to be processed and preserve multi-view face exemplars in a track. The detected face tracks are stored with the information of the face position, scale and the start and end frame number of the track. Then, we detect speaking face tracks among the face tracks. Finally, we cluster them into groups corresponding to characters and build the face affinity network.

A. Speaking Face Track Detection

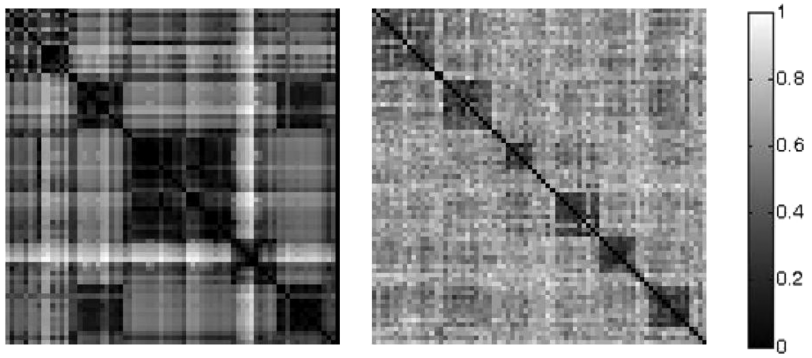
For speaking face track detection, we first determine the speaking face on the frame level in each face track. On each frame of the face track, the mouth region-of-interest (ROI) is located according to the face region. SIFT points are extracted and matched between the current face image and the previous one. Then, we use the matched SIFT points to calculate the transformation model to align the current face to the previous face image plane. The change in the aligned mouth ROI can be used to judge whether the face is speaking. Here, we use normalized sum of absolute difference (NSAD) [24] to describe the change in the mouth ROI. Thus, we get a vector of NSAD for each face track and use it to label the frame whether the face is speaking. If a face track has more than 10% frames labeled as speaking, it will be determined as a speaking face track. More technical details can be found in [24].

B. Face Representation by LLE

After face detection, each face is geometrically aligned into a standard form to remove the variation in translation, scale and in-plane rotation, and normalized into a 64×64 gray-scale image. Hence, each face can be represented as a 64×64 dimensional gray-scale feature vector. In our work, as multi-view or varied expressional faces of the same person should be considered as similar (i.e., to be treated as the neighboring points in an intrinsic manifold), it is required to employ a compact representation which can characterize this neighborhood relationship. Locally linear embedding (LLE) is such a nonlinear dimensionality reduction technique proposed by Roweis and Saul [25] which can map high dimensional data that are presumed to lie on a nonlinear manifold, onto a single global coordinate system of lower dimensionality, while still preserving the neighborhood relationship. LLE succeeds in recovering the underlying manifold, whereas linear embedding methods, such as PCA or multi-dimensional scaling (MDS), would map faraway data points to nearby points in their spaces. Hence, LLE is employed here to project high dimensional face features into the embedding space which can still preserve their intrinsic structures. The two panels of Fig. 2 show the first two components discovered by PCA and LLE. Note that while the linear projection by PCA has a somewhat uniform distribution, the LLE has a distinctly spiny structure.

C. Distance Measure Between Face Tracks

One face track is a set which may contain about 20–500 faces. Due to the variance of pose and expression, a face track may present multiple face exemplars. Matching different face tracks from the same person, just requires that certain faces of the two sets can be matched, while others are not necessarily as near as possible, but should also not be too far away in the feature space. Hence, it is necessary to find a distance measure which has the following properties: 1) it allows for partial matches, which is especially important in measuring between two sets with different sizes; 2) the severe dissimilarity will be punished. The face tracks from different persons may also have a few faces that look like the same due to the angle of view, illumination or image resolution. These partial similarities of the two sets



(a) (b)

Fig. 3. Distance matrices of face tracks from six characters measured by the (a) minimum distance and the (b) EMD. In the figure, the lower the intensity, the smaller the distance.

should be excluded by punishing the more severe dissimilarities of the other faces.

In most of the previous work [15], [18], minimum distances were employed to evaluate the dissimilarity of face tracks:

$$d(F_i, F_j) = \min_{f_m \in F_i} \min_{f_n \in F_j} \|f_m - f_n\| \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean distance; F_i and F_j are two face tracks; and f_m and f_n are the features of two faces belonging to them, respectively. The problem brought by minimum distance is that it only cares about the partial matching but does not punish the dissimilarities. In addition, the Euclidean distance used may not fit in certain situations. From panel (b) in Fig. 2, we can see that, although the LLE preserves the neighborhood relationship in the high dimensional space and has a distinctly spiny structure, their similarities still cannot be simply measured by Euclidean distance. For example, if using Euclidean norm, the distance between face track B and face track C is smaller than A and B, while actually A and B belong to the same person.

The EMD is a metric to evaluate the dissimilarity between two distributions [26]. It reflects the minimal amount of work that must be performed to transform one distribution into the other by moving "distribution mass" around. The EMD punishes the dissimilarity by increasing the amount of transportation work. It is represented as a distribution on certain dominant clusters, which are called *signature*. The signatures do not necessarily have the same mass, thus it allows for partial matches. Hence, the EMD is adequate to measure face track distance.

Here we come to face the key problem on extracting the dominant clusters. How can those varied faces which actually belong to one person be clustered into the same or near ground distance clusters? In panel (b) of Fig. 2, it is obvious that straightforwardly using the K-Means algorithm, whose resulting clusters are always convex sets, cannot work well. Here we employ spectral clustering [27] to do clustering on all the faces in the LLE space. The reason we choose spectral clustering is that it can preserve the local neighborhoods and solve very general problems like interwinds spirals. The number of clusters K is set by prior knowledge derived from the film script. This will be described in Section II-D in detail. We have compared the results of face clustering by spectral clustering and K-Means. The precision of spectral clustering is 69.6%, and the precision of K-Means is 58.1%. Since in building signatures, the dominant clusters are represented by their centers, it requires each cluster to be compact enough. We also calculate the mean intra-cluster distance of the clustering results. For spectral clustering, it is 3.41; for K-Means, it is 4.42. The experiments showed that spectral clustering is more suitable in our work. After spectral clustering, we can extract K dominant clusters from all the detected faces. Here it may be argued why we do not use majority voting within each face track to determine its cluster label. The reason is that majority voting cannot punish the dissimilarity either.

The face tracks can be represented as follows: Let $P = \{(c_{p_1}, w_{p_1}), \ldots, (c_{p_m}, w_{p_m})\}$ be the signature of the first face track with $m(m \leq K)$ clusters, where c_{p_i} is the cluster center and w_{p_i} is the number of faces belonging to this cluster; $Q = \{(c_{q_1}, w_{q_1}), \ldots (c_{q_n}, w_{q_n})\}$ be the signature of the second face track with $n(n \leq K)$ clusters. The EMD between two face tracks is defined as follows:

$$EMD(P, Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}} \tag{2}$$

where d_{ij} is the ground distance between cluster centers c_{p_i} and c_{q_j}. Note that the distance is calculated on the feature vectors of points derived from spectral clustering. The f_{ij} is the flow between c_{p_i} and c_{q_j}. The denominator of the equation is the normalization factor which is the total weight of the smaller signature. Calculation of the EMD is based on the solution of a linear programming problem subjecting to four constraints which can be found in [26]. To compare with the EMD, we also use the minimum distance to measure the distance between face tracks. The two panels of Fig. 3 illustrate the distance matrices of the face tracks measured by the (a) minimum distance and the (b) EMD, respectively. The face tracks are collected from one episode which contains six characters, and sorted by these characters. In panel (b), we can find six distinguished clusters. The intra-cluster distance is significantly smaller than the inter-cluster distance. In panel (a), although the intra-cluster distance is small, some face tracks from different characters also have small distances. The minimum distance makes them be treated as the same person due to the partial similarities.

D. Constrained K-Means Clustering

After computing the EMD between face tracks, a constrained K-Means clustering is performed to group the scattered face tracks which belong to the same character. Here, to exploit the properties of video, the temporal overlapping of the face tracks is implemented as a "cannot link" constraint when clustering: the two face tracks which share the common frames cannot be clustered together. The target number of clusters on face tracks is the same as K we set in spectral clustering on the faces. K is determined as follows: based on observation, we found that most of the speeches accompany the appearances of the faces in the video. Hence, we count the number of distinct speaker names appearing in the script and set K as this number. Here the "voice-over" or "off screen voice" in the films is not considered because they are labeled as "V. O." or "O. S." in the script and can be excluded by preprocess. We also ignore those characters whose spoken lines are less than three in the script, because these minor supporting roles appear in limited time and their face tracks can be considered as noise contrasting to the huge face track amounts of the others. To clean the noise from the clustering results, a pruning method is employed in the next step.

E. Cluster Pruning

In this step, we refine the clustering results by pruning the marginal points which have low confidence belonging to the current cluster. The confidence is calculated as follows:

$$\mathcal{C}(F) = \frac{k_{in}}{k} \cdot e^{-\mathcal{D}(F,F_0)} \tag{3}$$

where $\mathcal{D}(F, F_0)$ is the EMD between the face track F and its cluster center F_0; k is the number of K-nearest neighbors of F; and k_{in} is the number of K-nearest neighbors which belong to the same cluster with F. The point whose confidence is lower than a threshold Th_{conf} is regarded as the marginal point.

We collect all the marginal points pruned from the clusters and do a re-classification which incorporates the speaker voice features for enhancement. The reason that we do not combine the speaker voice features with the face features earlier in the face track clustering is that sometimes the environment or background sounds in the films are noisy. Directly fusing the speaker voice feature and the facial feature may affect the clustering result. For testing, in the face track clustering step, we concatenate the speaker voice feature and the facial feature into one feature vector to generate face track clusters. The precision of the clustering result is 64.3%, while the result of using facial feature only is 72.1%. It showed that the early feature fusion degrades the clustering result. Hence, we employ the speaker voice features to only reclassify the marginal points which are not confident by the facial features. As the marginal points are all speaking face tracks and the speaking frames have been detected in Section II-A, we can obtain speech data of each face track by segmenting corresponding clips from the film audio track. For each face track cluster, 30-s speech data are collected 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

$$p(\vec{x}|\lambda) = \sum_{i=1}^{M} w_i p_i(\vec{x}) \tag{4}$$

where $p_i(\vec{x})$ is the ith unimodal Gaussian densities; the mixture weights satisfy the constraint $\sum_{i=1}^{M} w_i = 1$; and $\vec{x}$ is a 24-dimensional Mel-frequency cepstral coefficients (MFCCs) feature vector. We remove DC-mean of the features and normalize the features by their cepstral mean in considering the background noise in the film. Under the assumption of independent feature vectors, the likelihood of a model λ for a sequence of feature vectors $X = \{\vec{x_1}, \ldots, \vec{x_T}\}$ is computed as follows:

$$\mathcal{P}(X|\lambda) = \prod_{t} p(\vec{x_t}|\lambda) \tag{5}$$

where $p(\vec{x_t}|\lambda)$ is computed as in (4). The speaker whose model gives the maximum likelihood is determined as the target speaker.

For classification, as we have learned discriminative functions from face and voice features, we adopt late fusion [28] to combine these results and yield a final classification score. Let C_k be the kth face track cluster whose cluster center is F_{C_k}, and the corresponding speech voice model is λ_{C_k}. Let F be the face track to be classified and X be the feature vector of the corresponding audio clip. The final discriminative function is defined as follows:

$$\mathcal{S}(F, C_k) = \alpha \cdot e^{-\mathcal{D}(F,F_{C_k})} + \beta \cdot \mathcal{P}(X|\lambda_{C_k}) \tag{6}$$

where $\mathcal{D}(F, F_{C_k})$ is the EMD between F and the cluster center F_{C_k}; $\mathcal{P}(X|\lambda_k)$ is the likelihood of C_k's voice model λ_{C_k} for X; and α and β are set as 0.4 and 0.6, respectively. The face track will be classified into the cluster whose function score $\mathcal{S}(F, C_k)$ is maximal. As described in Section II-D, there exist some face tracks belonging to the characters we have ignored. To clean these noises, we set a threshold Th_{score}. If the function score is lower than Th_{score}, the face track is refused to classify to any of the clusters and will be left unlabeled.

III. Face-Name Association

In the video and in the film script, the faces and the names can both stand for the characters. By treating all the characters as a small society, we can, respectively, build a name affinity network and a face affinity network in their own domains (script and video). For face-name association, we want to seek a matching between the two networks. From the social network point of view, our work is to find the structural equivalence actors between the two networks. Two actors, respectively, from the two networks are defined to be structural equivalence if they have the same profile of relationship to other actors in their own networks.

A. Name Affinity Network Building

A film script contains the spoken lines of characters together with the scene information and some brief descriptions. Fig. 4

INT. KITCHEN – DAY
William is tidying up frantically. Then he hears Anna's feet on the stairs.
She walks down, wearing a short, sparkling black top beneath her jacket.
He is dazzled by the sight of her.
WILLIAM
Would you like a cup of tea before you go?
ANNA
No thanks.
WILLIAM
Coffee?
ANNA
No.
WILLIAM
Orange juice -- probably not.
He moves to his very empty fridge -- and offers its only contents.

Fig. 4. Script examples of the film "Notting Hill".

shows a part of the script of the film "Notting Hill". The first line is the scene title (i.e., "INT. KITCHEN—DAY"), which has a standard format in the text. "INT." means interior, while its opposite "EXT." means exterior. "DAY" and "NIGHT" (in other scene titles) indicate the scene time. Following the scene title, there are some descriptions on the environment and the actions of characters. In front of each spoken line, there is the speaker name. Hence, we can parse the script text and use a name entity recognition software[1] to extract every name in front of the spoken lines. By recognizing the scene titles, we can compute the name occurrence counts of every character within each scene. The status of name occurrence in every scene can be formulated as an occurrence matrix $O_{name} = [o_{ik}]_{m\times n}$, where m is the number of names and n is the number of scenes. The entry o_{ik} of the matrix denotes the name count of the ith character in the kth scene. The ith row vector $\mathbf{o}_i = \{o_{i1}, o_{i2}, \ldots, o_{im}\}$ denotes the occurrence status of the ith character in all the scenes.

Based on the name occurrence matrix O_{name}, the name affinity network which is presented by a matrix $R_{name} = [r_{ij}]_{m\times m}$ can be constructed. The affinity value between two names is represented by their co-occurrence. The co-occurrence between name i and j in the kth scene is defined as follows:

$$c_{ijk} = \min(o_{ik}, o_{jk}). \tag{7}$$

Hence, the entry r_{ij} in the matrix R_{name} which represents the affinity value between name i and j in the entire script can be computed as follows:

$$r_{ij} = \sum_{k=1}^{n} c_{ijk} = \sum_{k=1}^{n} \min(o_{ik}, o_{jk}). \tag{8}$$

The diagonal value r_{ii} of the matrix is the occurrence count of the ith name in the entire script. Table I demonstrates the name affinity matrix of some true names choosing from the script of the film "Notting Hill". All the values are normalized into the interval [0,1].

[1]http://www.alias-i.com/lingpipe

TABLE I
EXAMPLE OF NAME AFFINITY MATRIX IN "NOTTING HILL"

	WIL	SPI	ANN	MAX	BEL	HON
WILLIAM	0.173	0.024	0.129	0.009	0.013	0.008
SPIKE	0.024	0.017	0.007	0.001	0.002	0.003
ANNA	0.129	0.007	0.144	0.000	0.000	0.001
MAX	0.009	0.001	0.000	0.009	0.006	0.004
BELLA	0.013	0.002	0.000	0.006	0.011	0.006
HONEY	0.008	0.003	0.001	0.004	0.006	0.007

TABLE II
EXAMPLE OF FACE AFFINITY MATRIX IN "NOTTING HILL"

	Face1	Face2	Face3	Face4	Face5	Face6
Face1	0.186	0.011	0.130	0.008	0.014	0.013
Face2	0.011	0.012	0.005	0.000	0.001	0.002
Face3	0.130	0.005	0.157	0.000	0.000	0.001
Face4	0.008	0.000	0.000	0.005	0.004	0.001
Face5	0.014	0.001	0.000	0.004	0.006	0.003
Face6	0.013	0.002	0.001	0.001	0.003	0.006

B. Face Affinity Network Building

Supposing that we have obtained K face track clusters, we give each cluster a numerical label from 1 to K corresponding to the K unnamed characters. Same as in Section III-A, the face affinity network is also built based on face co-occurrence. Hence, we also need to get the face occurrence status of every cluster in each scene of the video.

Here we briefly introduce the video scene segmentation employed in our work. First, we detect the interlaced repetitive pattern of shots in the film. This pattern often occurs in the people conversation, in which the camera repetitively shoots from one speaker to the other. These interlaced repetitive shots are grouped into one shot. Then among the rest of the shots, the most visually similar adjacent ones are gradually merged together. The merging order of the shots is recorded as the discontinuity degree between the shots. The later the two shots are merged, the higher the degree of discontinuity they are. Hence, the scene segmentation points can be inserted in the boundary between two shots which have the high degree of discontinuity. The technical details can be found in [29]. By setting a discontinuity degree threshold T_d, we can obtain a case of scene partition. To align with the scene partition in the film script, we change T_d to get the same number of scenes in the video with the script.

Based on the scene segmentation results, we can compute the face occurrence matrix $O_{face} = [o_{ik}]_{m\times n}$ on each scene, where m is the number of faces, and n is the number of scenes. Here O_{face} has the same size with O_{name} because the number of face clusters is set the same as the number of distinct names in the script. Finally, the face affinity network which is represented by a matrix $R_{face} = [r_{ij}]_{m\times m}$ can also be constructed by following (8). Table II demonstrates the face affinity matrix of some face clusters derived from the video of the film "Notting Hill". All the values are normalized into the interval [0,1].

C. Vertices Matching Between Two Graphs

We have obtained the name affinity network R_{name} and the face affinity network R_{face}. They both can be represented as an undirected, weighted graph, respectively:

$$G_{name} = \langle V_n, E_n, W_n \rangle, \quad G_{face} = \langle V_f, E_f, W_f \rangle. \tag{9}$$

In G_{name}, the vertices $V_n = \{n_1, n_2, \ldots, n_m\}$ represent m names, the edges $E_n = \{e_{ij} | \, if \, (r_{ij}^{name} > 0)\}$ denote the relationship between vertices, the weights $W_n = \{r_{ij}^{name}\}$ denote the strength of the edge e_{ij}, and the weights $W_n = \{r_{ii}^{name}\}$ denote the self-feature of the vertex n_i (i.e., its occurrence feature). In G_{face}, the vertices $V_f = \{f_1, f_2, \ldots, f_m\}$ represent m faces, and the edges E_f and the weights W_f also represent the relationship between faces. Therefore, the face-name association problem can be formulated as a graph matching problem, which targets on finding the correct correspondence between the vertices of the two graphs. Note that this matching process should subject to the one-to-one constrain, due to the reason that one name can match at most one face and vice-versa.

1) Problem Formulation: Given two graphs G_{name}, containing m vertices, and G_{face}, also containing m vertices, there are $m \times m$ possible correspondence pairs (or *assignment*) $(n_i, f_{i'})$, where $n_i \in G_{name}$ and $f_{i'} \in G_{face}$. Our aim is to find a correct correspondence mapping $\{M : V_n \leftrightarrow V_f\}$. In the graph, each vertex has a weight which can be seen as its self-feature. In our case, it is the face or name occurrence feature. However, the occurrence feature is not discriminative enough to build correct correspondence between the vertices of the two graphs. From Tables I and II, we can see that some diagonal values are similar. Consequently, the relationship with other vertices should be taken into account and treated as the features of the current vertex. In an ideal situation, if the scene segmentation in the video is as exact as the film script and the speaking face track clustering can achieve 100% precision, the name graph and the face graph should be exactly the same. The face graph can be seen as a transform from the name graph by adding noise which is introduced by speaking face track clustering and scene segmentation. Nevertheless, those two still reserve the relationship and the statistic properties of the characters, such as A has more affinities with B than with C, B never has co-occurrence with D, etc. Hence, we need to find a method using the relationship and statistic properties to build the correct correspondence which can accommodate a certain noise.

We store the $N(N = m \times m)$ candidate assignments in a list L. For each assignment $a = (n_i, f_{i'})$, we can find a measurement $M(a, a)$ on how well n_i matching $f_{i'}$:

$$M(a,a) = exp\left\{-\frac{\left(r_{ii}^{name} - r_{i'i'}^{face}\right)^2}{2\sigma^2}\right\} \tag{10}$$

where σ is the sensitivity parameter for accommodating noise or we can say the deformation between the two graphs. $M(a, a)$ can be seen as the individual feature of an assignment. A correct assignment often gets high value of $M(a, a)$.

For each pair of assignments (a, b), where $a = (n_i, f_{i'})$, $b = (n_j, f_{j'})$, we can also find a measurement $M(a, b)$ on how compatible the two assignments are. For example, on one hand, f_i and f_j have an affinity r_{ij}^{face}; on the other hand, $n_{i'}$ and $n_{j'}$ also have an affinity $r_{i'j'}^{name}$. If the pair of assignments (a, b) are both correct, the affinity values r_{ij}^{face} and $r_{i'j'}^{name}$ should be similar. Hence, $M(a, b)$ is defined as follows:

$$M(a,b) = exp\left\{-\frac{\left(r_{ij}^{name} - r_{i'j'}^{face}\right)^2}{2\sigma^2}\right\}. \tag{11}$$

$M(a, b)$ can be seen as the pairwise feature of two assignments. A pair of correct assignments are probably to agree with each other and get high value of $M(a, b)$. Based on this definition, $M(a, b)$ is nonnegative and symmetric $(M(a, b) = M(b, a))$. If two assignments are incompatible to the one-to-one matching constraint [i.e., $a = (n_i, f_{i'})$, $b = (n_i, f_{j'})$], we set $M(a, b) = 0$. Now the correspondence problem is reduced to finding a cluster C of assignments $(n_i, f_{i'})$ that maximizes the intra-cluster score while meeting the one-to-one matching constraint. The intra-cluster score is given as follows:

$$S = \sum_{a,b \in C} M(a,b) + \sum_{a \in C} M(a,a). \tag{12}$$

2) Spectral Matching Method: Spectral methods are commonly used for finding the main clusters of a graph. A spectral technique was introduced by Leordeanu and Hebert [30] for correspondence problem using pairwise constraints. They build an affinity matrix $M_{n \times n}$ of a graph whose vertices represent the potential correspondences and the weights on the edges represent pairwise agreements between potential correspondences. To find the cluster C which has the maximal inter-cluster score [see (12)], they define an indicator vector $x \in R^n$, where its element $x(i)$ is the confidence of the ith assignment a_i belonging to cluster C. The norm of x is fixed to 1. They aim to get the optimal solution x^*, where $x^* = arg_x \max(x^T M x)$. As we know, $M_{n \times n}$ is a symmetric and nonnegative matrix. By the Rayleigh quotient theorem, $x^T M x$ will be maximized when x is the principal eigenvector x^* of M. Since $M_{n \times n}$ has nonnegative elements, by the Perron–Frobenius theorem, the elements of x^* will be in the interval [0,1]. Hence, we can calculate the principal eigenvector x^* to determine the correct correspondences.

Inspiring from the method in [30], we first initialize the list L with the set of $m \times m$ possible assignments. Then we use the individual feature $M(a, a)$ and pairwise feature $M(a, b)$ defined above to build the affinity matrix $M_{m^2 \times m^2}$ which contains all possible assignments and is symmetric and nonnegative. From M, the principal eigenvector x^* can be calculated. We start by first accepting the most correct assignment a^* whose eigenvector value $x^*(a^*)$ is maximum. Next we reject all other assignments which are in conflict with a^* subjecting to the one-to-one matching constrain. Then we accept the next most correct assignment and reject the ones in conflict with it. This procedure will be repeated until all assignments are either accepted or rejected. The accepted assignments are the final results of name-face association.

3) Spectral Matching With Priors: The method introduced above is conducted in a totally unsupervised fashion. However, sometimes we can have certain prior knowledge such as we have known a correct assignment of a name and a face beforehand. The question is whether we can get benefit from such kind of priors on spectral matching. The known assignment can be obtained by the alignment of the film script and the subtitle using the method described in [18]. Although the global matching method we proposed in this paper does not need the timing information from the subtitle to generate local name cues, we want to investigate, if they are available, whether the local name cues can improve the global matching result. We first obtain the subtitle text from the video by OCR and then align the film script with the subtitle text by a dynamic time warping algorithm [31].

The result is that each script line is tagged with time stamps from the subtitle. Then the speaking face tracks are labeled with the names which have the corresponding time stamps. In each face track cluster which we have built in Section II, certain tracks are labeled with a name. Due to the errors of the two text sources alignment and the imprecise of speaking face track detection, some tracks may be mislabeled with wrong names. However, there is no mechanism for error correction in [18]. The face tracks from the same character may be labeled with different names. Hence, for each face track cluster, it should be assigned the majority name. The probability that the cluster C_{face} is assigned the majority name n^* is defined as

$$P(n^*|C_{face}) = \max_{n_i} \frac{N_{n_i}}{N} \tag{13}$$

where N_{n_i} is the number of face tracks which are assigned the name n_i in the cluster C_{face}, and N is the total number of face tracks in the cluster C_{face}. To obtain the most believable assignment of face and name, we select the one which has the highest probability value, and consider it as the known assignment.

Given a known assignment $a_0 = (n_0, f_0)$, where $M(a_0, a_0) = 1$, the relationship of other assignments a_i with a_0 [i.e., $M(a_0, a_i)$] is therefore more reliable. For a name $n_i (i \neq 0)$, it has a set of $m-1$ possible assignments: $S_a = \{a|(n_i, f_1), (n_i, f_2), \ldots, (n_i, f_{m-1})\}$, and thus has a set of $(m-1)$ pairwise features with a_0: $S_M = \{M|M(a_0, a_i), i \in [1, m-1]\}$. Among S_M, we use a "The Best Takes All" operation:

$$M(a_0, a_i) = \begin{cases} 1, & \text{if } a_i = arg\max_i M(a_0, a_i) \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

This operation will be done for all the names $n_i (i \neq 0)$. According to the symmetry characteristic, $M(a_i, a_0)$ will be set as the same value as $M(a_0, a_i)$. In addition, since $a_0 = (n_0, f_0)$ is the known correct assignment, subjecting to the one-to-one matching constraint, those assignments of the form $a_{0*} = (n_0, f_*)$ or $a_{*0} = (n_*, f_0)$ are in conflict with it. In the matrix M, the entries $M(a_{0*}, *)$ or $M(*, a_{0*})$ corresponding to these conflicting assignments are all set to 0. After all of these operations, the affinity matrix M will incorporate the prior knowledge and be transformed to be more sparse. Experimental results showed that incorporating priors can facilitate the graph matching.

IV. Applications

Until now, we have associated a name to each speaking face track cluster; thus, all the speaking face tracks can be identified. For the rest of the non-speaking face tracks we have detected before, we can also classify them into the nearest speaking face track clusters depending on the EMD defined in (2), and associate names to them. Note that this classification process is only relied on facial features.

Based on the result of character identification, there are many applications, such as character-based video retrieval, personalized video summarization, intelligent playback and video semantic mining, etc. Here we provide a platform for character-centered film browsing on which users can use character names to search and digest the film content.

A. Character Relationship Mining

To facilitate the character-centered browsing, the character relationship is mined first, which includes the determination of leading characters and cliques. Since the name affinity network and the face affinity network can both describe the relationship of characters and the name affinity network is more accurate, the relationship mining is conducted on the name affinity network.

From the social network analysis point of view, the leading character can be considered as the one who has high centrality in the name affinity network $R_{name} = [r_{ij}]_{m \times m}$. The centrality c_i of a character x_i is defined as $c_i = \sum_{j \neq i} r_{ij}$. Then the leading characters can be determined using the method of detecting the centrality gap among the characters [21]. We sort the centralities of characters in a descending order: $\{c_{1'} > c_{2'} > \ldots > c_{m'}\}$. Then we calculate the centrality difference $d_{i',i'+1}$ between two adjacent ones. The maximum difference $d_{i'^*,i'^*+1}$ will be set as the centrality gap which can distinguish the leading characters and the others. The ones whose centrality is $c_{i'} (i' \leq i'^*)$ are determined as the leading ones.

Clique is a subset of a network in which the actors are more closely and intensely tied to one another than they are to other members of the network. For clique detection, we use agglomerative hierarchical clustering [32]. The m individuals x_i are first initialized as m cliques. An empty clique List L is also located. The major steps are contained in the following procedure:

ALGORITHM (*Agglomerative Hierarchical Clustering*)

1) Begin initialize: $N \leftarrow m$, $C_i^0 \leftarrow \{x_i\}$, $i = 1, \ldots, m$,
2) $\theta \leftarrow Th$, $L = \varnothing$
3) **do** $N \leftarrow N - 1$
4) find nearest cliques, say C_k^t and C_l^t
5) **if** $S(C_k^t, C_l^t) > \theta$
6) **then** $C_k^{t+1} \leftarrow C_k^t \cup C_l^t$, $L \leftarrow C_k^{t+1}$
7) **else** break
8) **until** $N = 2$
9) **return** L

where $S(C_k^t, C_l^t)$ is defined as follows:

$$S\left(C_k^t, C_l^t\right) = \frac{\sum_{i \in C_k^t} \sum_{j \in C_l^t} r_{ij}}{||C_k^t|| \cdot ||C_l^t||}. \tag{15}$$

$||C_k^t||$ and $||C_l^t||$ are the numbers of the characters in C_k^t and C_l^t. In each step, the new merged clique is saved into the list L. We classify the result cliques into dyad which has two members, triad which has three members and the large clique. They will be listed in a summary of the film for character-centered browsing.

B. Character-Centered Browsing

Now we will provide a platform to support character-centered film browsing. We have identified the faces of characters. Hence, we can use character names to annotate the scenes in

TABLE III
EXAMPLES OF FILM CHARACTER SUMMARY

Film name:
Notting Hill
Leading characters:
1 William, 2 Anna
Others:
1 Spike, 2 Max, 3 Bella, 4 Honey, 5 Bernie, 6 Martin, 7 Jeremy, 8 Karen, 9 Thief, 10 Tarquin, 11 Jeff
Dyads:
1{William, Anna}, 2{Max, Bella}
Triads:
1{William, Anna, Spike}, 2{Max, Bella, Bernie}, 3{Jeremy, Karen, Tarquin}
Large cliques:
1{William, Anna, Spike, Honey}, 2{William, Anna, Spike, Honey, Max, Bella, Bernie}

TABLE IV
QUERY EXAMPLES

Query Example	Name	Ordinal
all the scenes of William	William	all
1st scene of William and Anna	William, Anna	1
Max and Bella	Max, Bella	all
last scene of Triad 3	Triad 3	last
2nd scene of large clique 2	large clique 2	2
Thief and dyad 1	Thief, dyad 1	all

the video. The annotation structure of one scene is defined as follows:

$$\langle Character\rangle(name_1, name_2, \ldots)$$
$$\langle Clique\rangle(clique_1, clique_2, \ldots).$$

Users can use the names of characters or cliques in the query to view the related video scenes. For the convenience of users, a summary on characters of the film is listed automatically which contains the characters (lead and others) and cliques. Taking the film "Notting Hill" as an example, the summary is shown in Table III.

Based on the summary, the query can be represented by using a short sentence, e.g., "1st scene of Anna", or "All scenes of William". For each query, we need to extract the keywords to infer the intents of the users. The query Q is formulated as follows:

$$Q = Name + Ordinal \tag{16}$$

where $Name = \{Character\ name, Clique\ name\}$, $Ordinal = \{1, 2, \ldots, last, all\}$. If there is no ordinal number in the query, the default value is all. Some examples of queries are given in Table IV.

V. EXPERIMENT

To evaluate our character identification approach, the experiments are conducted on ten feature-length films: "Notting Hill", "Pretty Woman", "Sleepless in Seattle", "You've Got Mail", "Devil Wears Prada", "Legally Blond", "Revolutionary Road", "The Shawshank Redemption", "Léon", and "Mission: Impossible". The information of these films are shown in Table V.

TABLE V
FILM INFORMATION

ID	Film	Length	Genres
F1	Notting	124 min	Comedy/Drama/Romance
F2	Pretty	119min	Comedy/Drama/Romance
F3	Seattle	105 min	Comedy/Drama/Romance
F4	Mail	119 min	Comedy/Romance
F5	Prada	109min	Comedy/Drama
F6	Blond	96min	Comedy
F7	Revolutionary	119min	Drama/Romance
F8	Shawshank	142 min	Crime/Drama
F9	Léon	110min	Crime/Thriller
F10	Mission	110 min	Action/Adventure/Thriller

TABLE VI
SPEAKING FACE TRACK DETECTION

Clip	Face trk	Sp. trk	Sp. detect	Accuracy
1	507	423	405	95.7%
2	459	372	348	93.5%
3	586	539	513	95.2%

A. Face Clustering

As the preliminary, we start with the speaking face track detection. To assess its accuracy, we segment three 30-min clips, respectively, from the films F1, F3, and F8. The statistics of the experiment are shown in Table VI, where the columns "Face trk" and "Sp. trk" are the total numbers of face tracks and speaking face tracks contained in the clips.

After speaking face track detection, we cluster them into groups corresponding to the characters. For each cluster, the cluster pruning mechanism is then used to refine the results. The point whose confidence $\mathcal{C}$ [see (3)] is lower than the confidence threshold Th_{conf} is determined as the marginal point and pruned. Hence, Th_{conf} is the parameter to control the purity of each cluster. The higher the value of Th_{conf} is, the more points will be pruned. To demonstrate the results of our method on face track clustering, we change the value of Th_{conf} from 0.5 to 0.1 and obtain a clustering precision/recall curve (see Fig. 5). The term "recall" is used here to indicate the proportion of the points not pruned against the total points. The calculation of precision and recall are given as follows. To avoid possible confusion with the traditional definition, we use "*" for distinguishing:

$$precision^* = \frac{points\ correctly\ clustered}{total\ points - marginal\ points} \tag{17}$$
$$recall^* = \frac{total\ points - marginal\ points}{total\ points}. \tag{18}$$

For comparison, we also use the minimum distance measure instead of our method during clustering. The result is shown in Fig. 5. The result of spectral clustering on faces before applying EMD in Section II-C is also shown as the baseline. It can be seen that our method is more effective to characterize the similarity between face tracks and get better clustering results.

After the cluster pruning, we collect the pruned marginal points and do a re-classification. As these are all speaking face tracks, the speaker voice features are fused with the facial features for classification. In Section II-E, we have set a score threshold Th_{score} to discard noise. By changing the value of

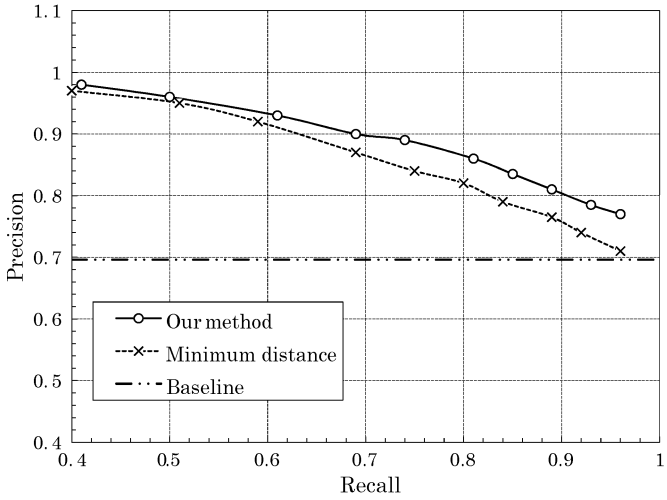


Fig. 5. Precision/recall curves of face track clustering.

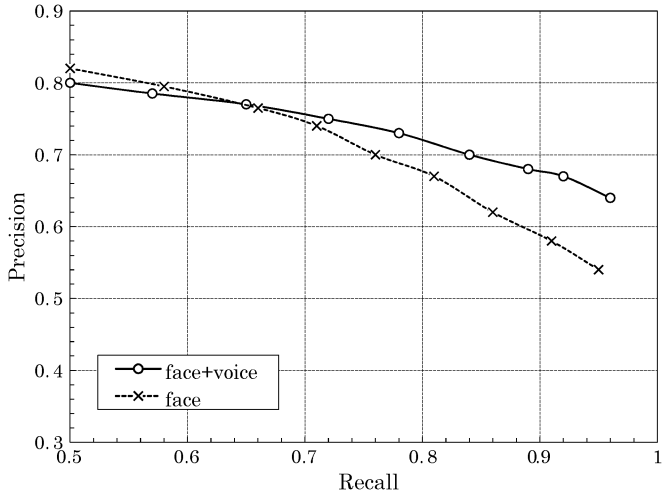


Fig. 6. Precision/recall curves of face track classification.

Th_{score} from 0.75 to 0.1, we can also get a precision/recall curve for face track classification (see Fig. 6). Similarly, here the term "recall" means the proportion of face tracks which are classified. The calculation of the precision and recall are given as follows. Also, we use "*" to distinguish from the traditional definition:

$$precision^* = \frac{facetracks\ correctly\ classified}{facetracks\ classified} \quad (19)$$

$$recall^* = \frac{facetracks\ classified}{total\ facetracks}. \quad (20)$$

Before face-name association stage, our work is only conducted on speaking face tracks. Thus, after face-name association, we also classify the non-speaking face tracks into the clusters we have built. The classification of non-speaking face tracks is based on facial features only. The results are also demonstrated in Fig. 6. As expected, the performance of multi-modal features (face + voice) is better than single feature (face) on the marginal points.

TABLE VII
NAME-FACE ASSOCIATION

Film	No. of characters	Correct named	Accuracy
F1	13	13	100 %
F2	13	11	84.6 %
F3	21	17	81.0 %
F4	14	12	85.7 %
F5	12	10	83.3 %
F6	20	16	80.0 %
F7	13	11	84.6 %
F8	14	12	85.7 %
F9	14	10	71.4 %
F10	17	13	76.5 %

TABLE VIII
NAME-FACE ASSOCIATION WITH PRIOR

Film	No. of characters	Correct named	Accuracy
F1	13	13	100 %
F2	13	11	84.6 %
F3	21	19	90.5 %
F4	14	14	100 %
F5	12	12	100 %
F6	20	18	90.0 %
F7	13	11	84.6 %
F8	14	12	85.7 %
F9	14	12	85.7 %
F10	17	15	88.2 %

B. Face-Name Association

We have obtained different clusters of face tracks corresponding to different characters. For assigning names to these clusters, a spectral matching method is employed to achieve vertices matching between name and face networks. The results on the ten films are shown in Table VII. It can be seen that the accuracy of the thriller and action film (F9 and F10) is lower than others. It is due to the more severe variation of the face pose and the illumination in the thriller and action films. In F9, the characters sometimes wear masks. In a scene of F10, the hero even disguises his face as the other character. These matters affect the face clustering and bring noise in the face affinity matrix. Thus, more errors occur in face-name association. Since the proposed method can incorporate priors to improve the matching, we give one known assignment of a name and a face, which is generated in Section III-C3, as a prior for each film. The results (see Table VIII) validate the effectiveness of adding priors in the matching process.

A comparison with the existing local matching approach was carried out. The approach [18] proposed by Everingham *et al.* was evaluated on the same dataset. We implemented the approach strictly obeying the original description in [18]. The alignment of the film script and the subtitle had been implemented in Section III-C3 to obtain local name cues. The speaking face tracks were then labeled with a temporally local name and set as exemplars. Other face tracks were classified to these exemplars for labeling. A precision/recall curve was obtained to demonstrate the performances. The term "recall" means the proportion of tracks which are assigned a name, and "precision" is the proportion of correctly labeled tracks [18]. To compare with this approach, we use the names assigned to the clusters to label the face tracks in the clusters. Similarly, we also obtain a precision/recall curve by changing the score threshold

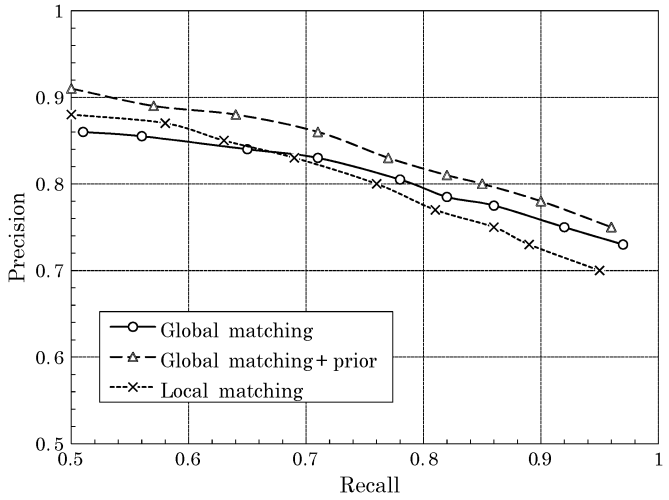


Fig. 7. Precision/recall curves of character identification.

TABLE IX
RELATIONSHIP MINING

Film	No. of leads	Detect	No. of cliques	Detect
F1	2	2	9	7
F2	2	2	8	7
F3	3	3	11	8
F4	2	2	9	8
F5	2	1	8	6
F6	1	1	12	9
F7	2	2	7	7
F8	2	2	6	5
F9	2	2	6	4
F10	1	1	10	7

TABLE X
USER EVALUATION OF CHARACTER-CENTERED BROWSING

	1	2	3	4	5	Average
Completeness	0	0	1	7	2	4.1
Acceptance	0	0	0	4	6	4.6
Novelty	0	0	0	2	8	4.8

Th_{score} defined in Section II-E to discard some face tracks without labeling. To demonstrate the improvement brought by the prior knowledge introduced in face name matching, the character identification results with the prior are also illustrated as a precision/recall curve. The three curves are shown in Fig. 7. It can be seen that the global matching method we proposed is comparable to the local matching method [18], while using less information source (film script) than it (film script + subtitle). At the high levels of recall, our method even performs better. This mainly relies on the effectiveness of the face track distance measure in clustering and the employment of the multi-modal features in cluster pruning. From the curve of "Global matching + prior", we can also find that incorporating certain local name cues as the prior knowledge does improve the character identification results, though our method actually does not rely on it. Since our method only needs the film script as the text information source, it can be applied under the circumstances that not enough time information can be found. In addition, the method in [18] was restricted to the frontal faces, while our method deals with the multi-view faces.

C. Relationship Mining

The performance of relationship mining is shown in Table IX. The columns "No. of leads" and "No. of cliques" are manually labeled ground truth. As this process is conducted on the name affinity network which is derived from the film script, the social network analysis on this clean data performs well. A few cliques are not detected due to the reason that besides the co-occurrence, more semantical information is needed to detect them. For example, in F3, the hero and the heroine never meet each other until the last, but semantically they are considered as a clique as they fall in love with each other at last.

D. Character-Centered Browsing

Based on the results of character identification and relationship mining, we annotate the scenes with character names and clique names in the format we defined in Section IV-B. To evaluate the performance of the character-centered browsing, we invited ten subjects (six males and four females) to participate in the test. They are postgraduate students and research staff from 24 to 40 years old. They each were asked to use five queries in the form of the examples shown in Table IV to browse the related clips in the films. Then they each gave a score to the browsing results on three attributes: completeness, acceptance, and novelty. Completeness is to measure whether the user has watched what he/she wants. Acceptance is to measure how much the browsing style is accepted. Novelty is to measure whether it exceeds the browsing expectation of the user and brings him/her new experience. The score is based on the following scale: 5-very good, 4-good, 3-neutral, 2-bad, 1-very bad. The scores from all subjects are given in Table X. The results indicate that most of the users are interested in the character-centered browsing and accept this new browsing style. From the automatically generated character summary of the film, they can grasp the structure of characters in the film, and use it to select and digest the character-related contents. This provides a new alternative for film contents organization and summarization. One user suggested that the video annotation may be extended from scene level to shot level, which can make it more accurate and complete.

VI. CONCLUSIONS

In this paper, we have proposed a novel framework for character identification in feature-length films. Different from the previous work on naming faces in the videos, most of which relied on local matching, we have presented a global matching method. A graph matching method has been utilized to build name-face association between the name affinity network and the face affinity network which are, respectively, derived from their own domains (script and video). As an application, we have mined the relationship between characters and provided a platform for character-centered film browsing.

In the future, we will improve our current work along three directions. 1) In face-name association, some useful information such as gender and context information will be integrated to refine the matching result. 2) Currently film content search and browsing is on the scene level to keep the integrity of the story. We will extend video annotation and organization on the shot level to achieve better accuracy and completeness in responding

to the query of the user. 3) We will explore to generate a movie trailer related to a certain character or a group of characters.

ACKNOWLEDGMENT

The authors would like to thank S. Chen, Y. Wu, and C. Zang for a number of helpful discussions and sharing necessary codes. The authors are also grateful to X.-Y. Chen for experimental data preparation and labeling.

REFERENCES

[1] W. Zhao, R. Chelappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Compu. Surv.*, vol. 35, no. 4, pp. 399–458, 2003.

[2] S. Satoh and T. Kanade, "Name-it: Association of face and name in video," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 1997, pp. 368–373.

[3] R. Houghton, "Named faces: Putting names to faces," *IEEE Intell. Syst.*, vol. 14, no. 5, pp. 45–50, 1999.

[4] J. Yang and A. G. Hauptmann, "Naming every individual in news video monologues," in *Proc. 12th Annu. ACM Int. Conf. Multimedia*, 2004, pp. 580–587.

[5] J. Yang, A. Hauptmann, and M.-Y. Chen, "Finding person x: Correlating names with visual appearances," in *Proc. Int. Conf. Image and Video Retrieval*, 2004, pp. 270–278.

[6] J. Yang, R. Yan, and A. G. Hauptmann, "Multiple instance learning for labeling faces in broadcasting news video," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, 2005, pp. 31–40.

[7] D. Ozkan and P. Duygulu, "Finding people frequently appearing in news," in *Proc. Int. Conf. Image and Video Retrieval*, 2006, pp. 173–182.

[8] T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y. Teh, E. Miller, and D. Foryth, "Names and faces in the news," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2004, vol. 2, pp. 848–854.

[9] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Automatic face naming with caption-based supervision," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2008.

[10] V. Jain, E. Learned-Miller, and A. McCallum, "People-LDA: Anchoring topics to people using face recognition," in *Proc. IEEE Int. Conf. Computer Vision*, 2007.

[11] O. Arandjelovic and A. Zisserman, "Automatic face recognition for film character retrieval in feature-length films," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2005, pp. 860–867.

[12] J. Sivic, M. Everingham, and A. Zisserman, "Person spotting: Video shot retrieval for face sets," in *Proc. Int. Conf. Image and Video Retrieval*, 2005, pp. 226–236.

[13] A. W. Fitzgibbon and A. Zisserman, "On affine invariant clustering and automatic cast listing in movies," in *Proc. Eur. Conf. Computer Vision*, 2002, vol. 3, pp. 304–320.

[14] O. Arandjelovic and R. Cipolla, "Automatic cast listing in feature-length films with anisotropic manifold space," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2006, pp. 1513–1520.

[15] Y. Gao *et al.*, "Cast indexing for videos by ncuts and page ranking," in *Proc. Int. Conf. Image and Video Retrieval*, 2007, pp. 441–447.

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

[17] Y. Li, S. Narayanan, and C.-C. J. Kuo, "Content-based movie analysis and indexing based on audiovisual cues," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 8, pp. 1073–1085, 2004.

[18] M. Everingham, J. Sivic, and A. Zisserman, ""Hello! My name is . . . Buffy" automatic naming of characters in TV video," in *Proc. British Machine Vision Conf.*, 2006, pp. 889–908.

[19] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar, "Movie/script: Alignment and parsing of video and text transcription," in *Proc. 10th Eur. Conf. Computer Vision*, 2008, pp. 158–171.

[20] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2008.

[21] C.-Y. Weng, W.-T. Chu, and J.-L. Wu, "Rolenet: Treat a movie as a small society," in *Proc. Int. Workshop Multimedia Information Retrieval*, 2007, pp. 51–60.

[22] J. Scott, *Social Network Analysis: A Handbook*. Newbury Park, CA: Sage, 1991.

[23] Y. Li, H. Z. Ai, C. Huang, and S. H. Lao, "Robust head tracking with particles based on multiple cues fusion," in *Proc. HCI/ECCV*, 2006, pp. 29–39.

[24] Y. Wu, W. Hu, T. Wang, Y. Zhang, J. Cheng, and H. Lu, "Robust speaking face identification for video analysis," in *Proc. Pacific Rim Conf. Multimedia*, 2007, pp. 665–674.

[25] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[26] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proc. IEEE Int. Conf. Computer Vision*, 1998, pp. 59–66.

[27] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Adv. Neural Inf. Process. Syst. 14*, pp. 849–856, 2001.

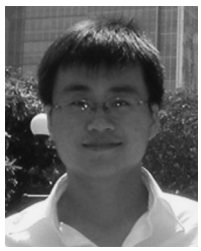
[28] C. Snoek, M. Worring, and A. Smeulders, "Early versus late fusion in semantic video analysis," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, 2005, pp. 399–402.

[29] T. Mei, X.-S. Hua, L. Yang, and S. Li, "Videosense—towards effective online video advertising," in *Proc. 15th Annu. ACM Int. Conf. Multimedia*, 2007, pp. 1075–1084.

[30] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *Proc. 10th IEEE Int. Conf. Computer Vision*, 2005, pp. 1482–1489.

[31] C. S. Myers and L. R. Rabiner, "A comparative study of several dynamic time-warping algorithms for connected word recognition," *Bell Syst. Tech. J.*, vol. 60, pp. 1389–1409, 1981.

[32] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York: Wiley, 2000.



Yi-Fan Zhang (S'09) received the B.E. degree from Southeast University, Nanjing, China, in 2004. He is currently pursuing the Ph.D. degree at National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

In 2007, he was an intern student in the Institute for Infocomm Research, Singapore. Currently he is an intern student in China-Singapore Institute of Digital Media. His research interests include multimedia, video analysis, and pattern recognition.

Changsheng Xu (M'97–SM'99) is a Professor in the Institute of Automation, Chinese Academy of Sciences, and Executive Director of China-Singapore Institute of Digital Media. He was with Institute for Infocomm Research, Singapore, from 1998 to 2008. He was with the National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, from 1996 to 1998. His research interests include multimedia content analysis, indexing and retrieval, digital watermarking, computer vision, and pattern recognition. He published over 170 papers in those areas.

Dr. Xu is an Associate Editor of *ACM/Springer Multimedia Systems Journal*. He served as Program Co-Chair of 2009 ACM Multimedia Conference, Short Paper Co-Chair of ACM Multimedia 2008, General Co-Chair of 2008 Pacific-Rim Conference on Multimedia and 2007 Asia-Pacific Workshop on Visual Information Processing (VIP2007), Program Co-Chair of VIP2006, Industry Track Chair, and Area Chair of 2007 International Conference on Multimedia Modeling. He also served as Technical Program Committee Member of major international multimedia conferences, including ACM Multimedia Conference, International Conference on Multimedia & Expo, Pacific-Rim Conference on Multimedia, and International Conference on Multimedia Modeling. He received the 2008 Best Editorial Member Award of *ACM/Springer Multimedia Systems Journal*. He is a member of ACM.

Hanqing Lu (M'05–SM'06) received the Ph.D. degree from Huazhong University of Sciences and Technology, Wuhan, China, in 1992.

Currently, he is a Professor in the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include image similarity measure, video analysis, object recognition, and tracking. He has published more than 100 papers in those areas.

Yueh-Min Huang (M'98) received the M.S. and Ph.D. degrees in electrical engineering from the University of Arizona, Tucson, in 1988 and 1991, respectively.

He is a Professor and Chairman of the Department of Engineering Science, National Cheng-Kung University, Tainan, Taiwan. His research interests include multimedia communications, wireless networks, artificial intelligence, and e-Learning. He has coauthored two books and has published about 200 refereed professional research papers.

Dr. Huang has received many research awards, such as the Best Paper Award of 2007 IEA/AIE Conference; the Awards of Acer Long-Term Prize in 1996, 1998, and 1999; and Excellent Research Awards of National Microcomputer and Communication Contests in 2006. He has been invited to give talks or served frequently in the program committee at national and international conferences. He is in the editorial board of the *Journal of Wireless Communications and Mobile Computing*, *Journal of Security and Communication Networks*, and *International Journal of Communication Systems*. He is a member of the IEEE Communication, Computer, and Circuits and Systems Societies.