

# Robust Face-Name Graph Matching for Movie Character Identification

Jitao Sang and Changsheng Xu, *Senior Member, IEEE*

**Abstract**—Automatic face identification of characters in movies has drawn significant research interests and led to many interesting applications. It is a challenging problem due to the huge variation in the appearance of each character. Although existing methods demonstrate promising results in clean environment, the performances are limited in complex movie scenes due to the noises generated during the face tracking and face clustering process. In this paper we present two schemes of global face-name matching based framework for robust character identification. The contributions of this work include the following. 1) A noise insensitive character relationship representation is incorporated. 2) We introduce an edit operation based graph matching algorithm. 3) Complex character changes are handled by simultaneously graph partition and graph matching. 4) Beyond existing character identification approaches, we further perform an in-depth sensitivity analysis by introducing two types of simulated noises. The proposed schemes demonstrate state-of-the-art performance on movie character identification in various genres of movies.

**Index Terms**—Character identification, graph edit, graph matching, graph partition, sensitivity analysis.

## I. INTRODUCTION

### A. Objective and Motivation

THE proliferation of movie and TV provides large amount of digital video data. This has led to the requirement of efficient and effective techniques for video content understanding and organization. Automatic video annotation is one of such key techniques. In this paper our focus is on annotating characters in the movie and TVs, which is called movie character identification [1]. The objective is to identify the faces of the characters in the video and label them with the corresponding names in the cast. The textual cues, like cast lists, scripts, subtitles and closed captions are usually exploited. Fig. 1 shows an example in our experiments. In a movie, characters are the focus center of interests for the audience. Their occurrences provide lots of clues about the movie structure and content. Automatic character identification is essential for semantic movie index and re-

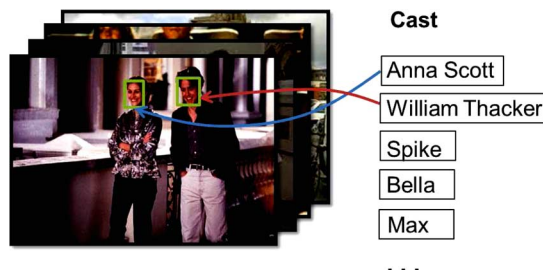


Fig. 1. Examples of character identification from movie “Notting Hill.”

trieval [2], [3], scene segmentation [4], summarization [5] and other applications [6].

Character identification, though very intuitive to humans, is a tremendously challenging task in computer vision. The reason is four-fold: 1) Weakly supervised textual cues [7]. There are ambiguity problem in establishing the correspondence between names and faces: ambiguity can arise from a reaction shot where the person speaking may not be shown in the frames<sup>1</sup>; ambiguity can also arise in partially labeled frames when there are multiple speakers in the same scene<sup>2</sup>. 2) Face identification in videos is more difficult than that in images [8]. Low resolution, occlusion, nonrigid deformations, large motion, complex background and other uncontrolled conditions make the results of face detection and tracking unreliable. In movies, the situation is even worse. This brings inevitable noises to the character identification. 3) The same character appears quite differently during the movie [3]. There may be huge pose, expression and illumination variation, wearing, clothing, even makeup and hairstyle changes. Moreover, characters in some movies go through different age stages, e.g., from youth to the old age. Sometimes, there will even be different actors playing different ages of the same character. 4) The determination for the number of identical faces is not trivial [2]. Due to the remarkable intraclass variance, the same character name will correspond to faces of huge variant appearances. It will be unreasonable to set the number of identical faces just according to the number of characters in the cast. Our study is motivated by these challenges and aims to find solutions for a robust framework for movie character identification.

### B. Related Work

The crux of the character identification problem is to exploit the relations between videos and the associated texts in order

Manuscript received September 30, 2011; revised January 20, 2012; accepted February 12, 2012. Date of publication February 22, 2012; date of current version May 11, 2012. This work was supported in part by the National Program on Key Basic Research Project (973 Program, Project No. 2012CB316304) and National Natural Science Foundation of China (Grant No. 90920303, 61003161). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sethuraman Panchanathan.

J. Sang and C. Xu 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, 119613 (e-mail: jtsang@nlpr.ia.ac.cn; csxu@nlpr.ia.ac.cn).

Digital Object Identifier 10.1109/TMM.2012.2188784

<sup>1</sup>I.e., the name in the subtitle/closed caption finds no corresponding faces in the video.

<sup>2</sup>I.e., multiple names in the subtitle/closed caption correspond to multiple faces in the video.

to label the faces of characters with names. It has similarities to identifying faces in news videos [9]–[11]. However, in news videos, candidate names for the faces are available from the simultaneously appearing captions or local transcripts. While in TV and movies, the names of characters are seldom directly shown in the subtitle or closed caption, and script/screenplay containing character names has no time stamps to align to the video. According to the utilized textual cues, we roughly divide the existing movie character identification methods into three categories.

1) *Category 1: Cast List Based:* These methods only utilize the case list textual resource. In the “cast list discovery” problem [12], [13], faces are clustered by appearance and faces of a particular character are expected to be collected in a few pure clusters. Names for the clusters are then manually selected from the cast list. Ramanan *et al.* proposed to manually label an initial set of face clusters and further cluster the rest face instances based on clothing within scenes [14]. In [15], the authors have addressed the problem of finding particular characters by building a model/classifier of the character’s appearance from user-provided training data. An interesting work combining character identification with web image retrieval is proposed in [17]. The character names in the cast are used as queries to search face images and constitute gallery set. The probe face tracks in the movie are then identified as one of the characters by multitask joint sparse representation and classification. Recently, metric learning is introduced into character identification in uncontrolled videos [16]. Cast-specific metrics are adapted to the people appearing in a particular video in an unsupervised manner. The clustering as well as identification performance are demonstrated to be improved. These cast list based methods are easy for understanding and implementation. However, without other textual cues, they either need manual labeling or guarantee no robust clustering and classification performance due to the large intraclass variances.

2) *Category 2: Subtitle or Closed Caption, Local Matching Based:* Subtitle and closed caption provide time-stamped dialogues, which can be exploited for alignment to the video frames. Everingham *et al.* [18], [3] proposed to combine the film script with the subtitle for local face-name matching. Time-stamped name annotation and face exemplars are generated. The rest of the faces were then classified into these exemplars for identification. They further extended their work in [19], by replacing the nearest neighbor classifier by multiple kernel learning for features combination. In the new framework, nonfrontal faces are handled and the coverage is extended. Researchers from University of Pennsylvania utilized the readily available time-stamped resource, the closed captions, which is demonstrated more reliable than OCR-based subtitles [20], [7]. They investigated on the ambiguity issues in the local alignment between video, screenplay and closed captions. A partially-supervised multiclass classification problem is formulated. Recently, they attempted to address the character identification problem without the use of screenplay [21]. The reference cues in the closed captions are employed as multiple instance constraints and face tracks grouping as well as face-name association are solved in a convex formulation. The local matching based methods require the time-stamped

information, which is either extracted by OCR (i.e., subtitle) or unavailable for the majority of movies and TV series (i.e., closed caption). Besides, the ambiguous and partial annotation makes local matching based methods more sensitive to the face detection and tracking noises.

3) *Category 3: Script/Screenplay, Global Matching Based:* Global matching based methods open the possibility of character identification without OCR-based subtitle or closed caption. Since it is not easy to get local name cues, the task of character identification is formulated as a global matching problem in [2], [22], [4]. Our method belongs to this category and can be considered as an extension to Zhang’s work [2]. In movies, the names of characters seldom directly appear in the subtitle, while the movie script which contains character names has no time information. Without the local time information, the task of character identification is formulated as a global matching problem between the faces detected from the video and the names extracted from the movie script. Compared with local matching, global statistics are used for name-face association, which enhances the robustness of the algorithms.

Our work differs from the existing research in three-fold:

- Regarding the fact that characters may show various appearances, the representation of character is often affected by the noise introduced by face tracking, face clustering and scene segmentation. Although extensive research efforts have been concentrated on character identification and many applications have been proposed, little work has focused on improving the robustness. We have observed in our investigations that some statistic properties are preserved in spite of these noises. Based on that, we propose a novel representation for character relationship and introduce a name-face matching method which can accommodate a certain noise.
- Face track clustering serves as an important step in movie character identification. In most of the existing methods, some cues are utilized to determine the number of target clusters prior to face clustering, e.g., in [2], the number of clusters is the same as the number of distinct speakers appearing in the script. While this seems convinced at first glance, it is rigid and even deteriorating the clustering results sometimes. In this paper, we loose the restriction of one face cluster corresponding to one character name. Face track clustering and face-name matching are jointly optimized and conducted in a unique framework.
- Sensitivity analysis is common in financial applications, risk analysis, signal processing and any area where models are developed [23], [24]. Good modeling practice requires that the modeler provides an evaluation of the confidence in the model, for example, assessing the uncertainties associated with the modeling process and with the outcome of the model itself. For movie character identification, sensitivity analysis offers valid tools for characterizing the robustness to noises for a model. To the best of our knowledge, there have been no efforts directed at the sensitivity analysis for movie character identification. In this paper, we aim to fill this gap by introducing two types of simulated noises.

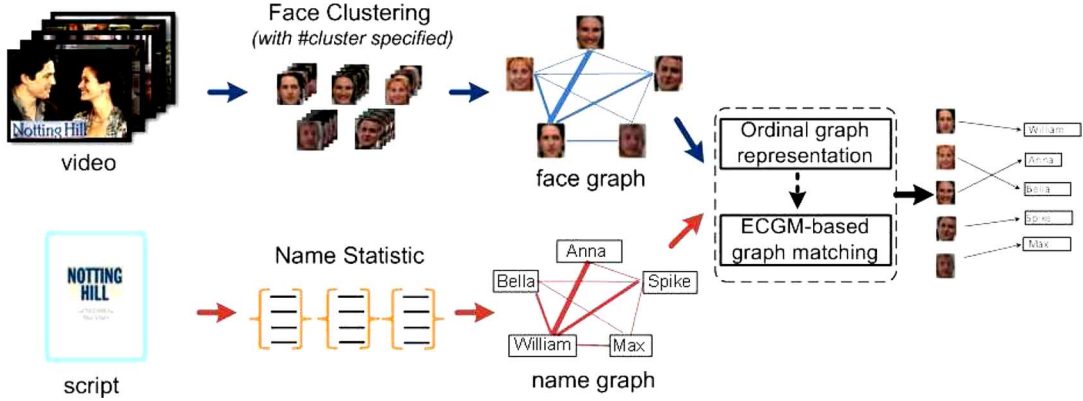


Fig. 2. Framework of scheme 1: Face-name graph matching with #cluster prespecified.

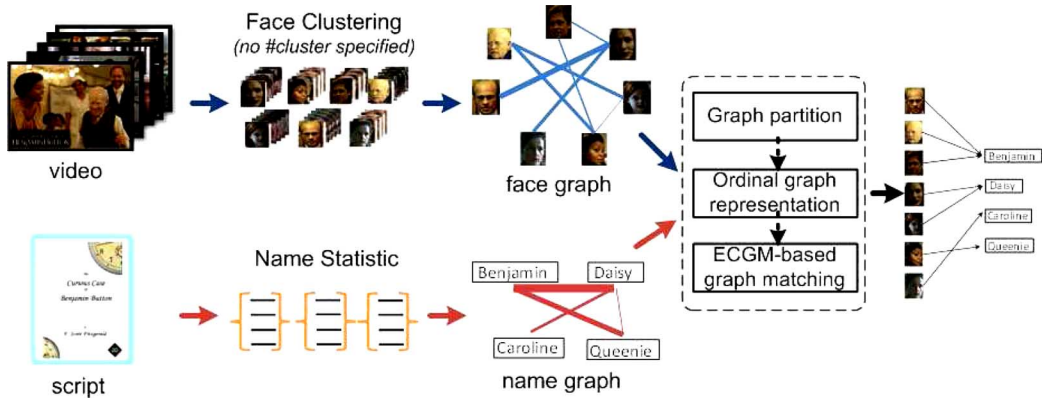


Fig. 3. Framework of scheme 2: Face-name graph matching without #cluster prespecified.

A preliminary version of this work was introduced by [1]. We provide additional algorithmic and computational details, and extend the framework considering no prespecification for the number of face clusters. Improved performance as well as robustness are demonstrated in movies with large character appearance changes.

### C. Overview of Our Approach

In this paper, we propose a global face-name graph matching based framework for robust movie character identification. Two schemes are considered. There are connections as well as differences between them. Regarding the connections, firstly, the proposed two schemes both belong to the global matching based category, where external script resources are utilized. Secondly, to improve the robustness, the ordinal graph is employed for face and name graph representation and a novel graph matching algorithm called Error Correcting Graph Matching (ECGM) is introduced. Regarding the differences, scheme 1 sets the number of clusters when performing face clustering (e.g., K-means, spectral clustering). The face graph is restricted to have identical number of vertexes with the name graph. While, in scheme 2, no cluster number is required and face tracks are clustered based on their intrinsic data structure (e.g., mean shift, affinity propagation). Moreover, as shown in Figs. 2 and 3, scheme 2 has an additional module of graph partition compared with scheme 1. From this perspective, scheme 2 can be seen as an extension to scheme 1.

1) *Scheme 1*: The proposed framework for scheme 1 is shown in Fig. 2. It is similar to the framework of [2]. Face tracks are clustered using constrained K-means, where the number of clusters is set as the number of distinct speakers. Co-occurrence of names in script and face clusters in video constitutes the corresponding face graph and name graph. We modify the traditional global matching framework by using ordinal graphs for robust representation and introducing an ECGM-based graph matching method.

For face and name graph construction, we propose to represent the character co-occurrence in rank ordinal level [25], which scores the strength of the relationships in a rank order from the weakest to strongest. Rank order data carry no numerical meaning and thus are less sensitive to the noises. The affinity graph used in the traditional global matching is interval measures of the co-occurrence relationship between characters. While continuous measures of the strength of relationship holds complete information, it is highly sensitive to noises.

For name-face graph matching, we utilize the ECGM algorithm. In ECGM, the difference between two graphs is measured by edit distance which is a sequence of graph edit operations. The optimal match is achieved with the least edit distance. According to the noise analysis, we define appropriate graph edit operations and adapt the distance functions to obtain improved name-face matching performance.

2) *Scheme 2*: The proposed framework for scheme 2 is shown in Fig. 3. It has two differences from scheme 1 in Fig. 2.

First, no cluster number is required for the face tracks clustering step. Second, since the face graph and name graph may have different number of vertexes, a graph partition component is added before ordinal graph representation.

The basic premise behind the scheme 2 is that appearances of the same character vary significantly and it is difficult to group them in a unique cluster. Take the movie “The Curious Case of Benjamin Button” for example. The hero and heroine go through a long time period from their childhood, youth, middle-age to the old-age. The intraclass variance is even larger than the interclass variance. In this case, simply enforcing the number of face clusters as the number of characters will disturb the clustering process. Instead of grouping face tracks of the same character into one cluster, face tracks from different characters may be grouped together.

In scheme 2, we utilize affinity propagation for the face tracks clustering. With each sample as the potential center of clusters, the face tracks are recursively clustered through appearance-based similarity transmit and propagation. High cluster purity with large number of clusters is expected. Since one character name may correspond to several face clusters, graph partition is introduced before graph matching. Which face clusters should be further grouped (i.e., divided into the same subgraph) is determined by whether the partitioned face graph achieves an optimal graph matching with the name graph. Actually, face clustering is divided into two steps: coarse clustering by appearance and further modification by script. Moreover, face clustering and graph matching are optimized simultaneously, which improve the robustness against errors and noises.

In general, the scheme 2 has two advantages over the scheme 1. (a) For scheme 2, no cluster number is required in advance and face tracks are clustered based on their intrinsic data structure. Therefore, the scheme 2 provides certain robustness to the intraclass variance, which is very common in movies where characters change appearance significantly or go through a long time period. (b) Regarding that movie cast cannot include pedestrians whose face is detected and added into the face track, restricting the number of face tracks clusters the same as that of name from movie cast will deteriorate the clustering process. In addition, there is some chance that movie cast does not cover all the characters. In this case, prespecification for the face clusters is risky: face tracks from different characters will be mixed together and graph matching tends to fail.

3) *Sensitivity Analysis*: Sensitivity analysis plays an important role in characterizing the uncertainties associated with a model. To explicitly analyze the algorithm’s sensitivity to noises, two types of noises, coverage noise and intensity noise, are introduced. Based on that, we perform sensitivity analysis by investigating the performance of name-face matching with respect to the simulated noises.

The rest of paper is organized as follows: We first introduce the ordinal graph representation and ECGM-based graph matching of scheme 1 in Section II. Section III presents the scheme 2 and the graph partition algorithm. In Section IV, we introduced two simulated noises for sensitivity analysis. A set of experiments with comparisons are conducted and discussed in Section V, and we conclude the paper in Section VI.

	WIL	SPI	ANN	MAX	BEL
WIL	0.173	0.024	0.129	0.009	0.013
SPI	0.024	0.017	0.007	0.001	0.002
ANN	0.129	0.007	0.144	0	0
MAX	0.009	0.001	0	0.009	0.006
BEL	0.013	0.002	0	0.006	0.011

(a)

	Face1	Face2	Face3	Face4	Face5
Face1	0.186	0.041	0.147	0.008	0.021
Face2	0.041	0.012	0.005	0.002	0.004
Face3	0.147	0.005	0.157	0	0.003
Face4	0.008	0.002	0	0.005	0.007
Face5	0.021	0.004	0.003	0.007	0.009

(b)

Fig. 4. Example of affinity matrices from movie “Notting Hill”: (a) Name affinity matrix  $R^{\text{name}}$  (b) Face affinity matrix  $R^{\text{face}}$ .

## II. SCHEME 1: FACE-NAME GRAPH MATCHING WITH NUMBER OF CLUSTER SPECIFIED

In this section we first briefly review the framework of traditional global graph matching based character identification. Based on investigations of the noises generated during the affinity graph construction process, we construct the name and face affinity graph in rank ordinal level and employ ECGM with specially designed edit cost function for face-name matching.

### A. Review of Global Face-Name Matching Framework

In a movie, the interactions among characters resemble them into a relationship network. Co-occurrence of names in script and faces in videos can represent such interactions. Affinity graph is built according to the co-occurrence status among characters, which can be represented as a weighted graph  $G = \{V, E\}$  where vertex  $V$  denotes the characters and edge  $E$  denotes relationships among them. The more scenes where two characters appear together, the closer they are, and the larger the edge weights between them are. In this sense, a name affinity graph from script analysis and a face affinity graph from video analysis can be constructed. Fig. 4 demonstrates the adjacency matrices corresponding to the name and face affinity graphs from the movie “Notting Hill”<sup>3</sup>. All the affinity values are normalized into the interval  $[0, 1]$ . We can see that some of the face affinity values differ much from the corresponding name affinity values (e.g.,  $\{WIL, SPI\}$  and  $\{Face1, Face2\}$ ,  $\{WIL, BEL\}$  and  $\{Face1, Face5\}$ ) due to the introduced noises. Subsequently, character identification is formulated as the problem of finding optimal vertex to vertex matching between two graphs. A spectral graph matching algorithm is applied to find the optimal name-face correspondence. More technical details can be referred to [2].

<sup>3</sup>The ground-truth mapping is  $WIL$ -Face1,  $SPI$ -Face2,  $ANN$ -Face3,  $MAX$ -Face4,  $BEL$ -Face5



	WIL	SPI	ANN	MAX	BEL
WIL	5	3	4	1	2
SPI	4	3	3	1	2
ANN	4	3	4	0	0
MAX	4	2	0	1	3
BEL	4	2	0	3	2

(a)

	Face1	Face2	Face3	Face4	Face5
Face1	5	3	4	1	2
Face2	4	3	3	1	2
Face3	4	3	4	0	2
Face4	4	2	0	1	3
Face5	4	2	1	3	2

(b)

Fig. 5. Example of ordinal affinity matrices corresponding to Fig. 4: (a) Name ordinal affinity matrix  $R^{\text{name}}$  (b) Face ordinal affinity matrix  $R^{\text{face}}$ .

### B. Ordinal Graph Representation

The name affinity graph and face affinity graph are built based on the co-occurrence relationship. Due to the imperfect face detection and tracking results, the face affinity graph can be seen as a transform from the name affinity graph by affixing noises. We have observed in our investigations that, in the generated affinity matrix some statistic properties of the characters are relatively stable and insensitive to the noises, such as character A has more affinities with character B than C, character D has never co-occurred with character A, etc. Delighted from this, we assume that while the absolute quantitative affinity values are changeable, the relative affinity relationships between characters (e.g., A is more closer to B than to C) and the qualitative affinity values (e.g., whether D has co-occurred with A) usually remain unchanged. In this paper, we utilize the preserved statistic properties and propose to represent the character co-occurrence in rank order.

We denote the original affinity matrix as  $R = \{r_{ij}\}_{N \times N}$ , where  $N$  is the number of characters. First we look at the cells along the main diagonal (e.g., A co-occur with A, B co-occur with B). We rank the diagonal affinity values  $r_{ii}$  in ascending order, then the corresponding diagonal cells  $\tilde{r}_{ii}$  in the rank ordinal affinity matrix  $\tilde{R}$

$$\tilde{r}_{ii} = I_{r_{ii}} \quad (1)$$

where  $I_{r_{ii}}$  is the rank index of original diagonal affinity value  $r_{ii}$ . Zero-cell represents that no co-occurrence relationship is specially considered, which is a qualitative measure. From the perspective of graph analysis, there is no edge between the vertexes of row and column for the zero-cell. Therefore, change of zero-cell involves with changing the graph structure or topology. To distinguish the zero-cell change, for each row in the original affinity matrix, we remain the zero-cell

unchanged. The number of zero-cells in the  $i$ th row is recorded as  $\text{null}_i$ . Other than the diagonal cell and zero-cell, we sort the rest affinity values in ascending order, i.e., for the  $i$ th row, the corresponding cells  $\tilde{r}_{ij}$  in the  $i$ th row of ordinal affinity matrix

$$\tilde{r}_{ij} = I_{r_{ij}} + \text{null}_i \quad (2)$$

where  $I_{r_{ij}}$  denotes the order of  $r_{ij}$ . Note that the zero-cells are not considered in sorting, but the number of zero-cells will be set as the initial rank order<sup>4</sup>. The ordinal matrix is not necessarily symmetric. The scales reflect variances in degree of intensity, but not necessarily equal differences. We illustrate in Fig. 5 an example of ordinal affinity matrices corresponding to the affinity matrices in Fig. 4. It is shown that although there are major differences between original name and face affinity matrices, the derived ordinal affinity matrices are basically the same. The differences are generated due to the changes of zero-cell. A rough conclusion is that the ordinal affinity matrix is less sensitive to the noises than the original affinity matrix. We will further validate the advantage of ordinal graph representation in the experiment section.

### C. ECGM-Based Graph Matching

ECGM is a powerful tool for graph matching with distorted inputs. It has various applications in pattern recognition and computer vision [26]. In order to measure the similarity of two graphs, graph edit operations are defined, such as the deletion, insertion and substitution of vertexes and edges. Each of these operations is further assigned a certain cost. The costs are application dependent and usually reflect the likelihood of graph distortions. The more likely a certain distortion is to occur, the smaller is its cost. Through error correcting graph matching, we can define appropriate graph edit operations according to the noise investigation and design the edit cost function to improve the performance.

For explanation convenience, we provide some notations and definitions taken from [28]. Let  $\mathcal{L}$  be a finite alphabet of labels for vertexes and edges.

*Notation:* A graph is a triple  $g = (\mathcal{V}, \alpha, \beta)$ , where  $\mathcal{V}$  is the finite set of vertexes,  $\alpha : \mathcal{V} \rightarrow \mathcal{L}$  is vertex labeling function, and  $\beta : \mathcal{E} \rightarrow \mathcal{L}$  is edge labeling function.

The set of edges  $\mathcal{E}$  is implicitly given by assuming that graphs are fully connected, i.e.,  $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ . For the notational convenience, node and edge labels come from the same alphabet<sup>5</sup>.

*Definition 1:* Let  $g_1 = (\mathcal{V}_1, \alpha_1, \beta_1)$  and  $g_2 = (\mathcal{V}_2, \alpha_2, \beta_2)$  be two graphs. An ECGM from  $g_1$  to  $g_2$  is a bijective function  $f : \hat{\mathcal{V}}_1 \rightarrow \hat{\mathcal{V}}_2$ , where  $\hat{\mathcal{V}}_1 \subseteq \mathcal{V}_1$  and  $\hat{\mathcal{V}}_2 \subseteq \mathcal{V}_2$ .

We say that vertex  $x \in \hat{\mathcal{V}}_1$  is substituted by vertex  $y \in \hat{\mathcal{V}}_2$  if  $f(x) = y$ . If  $\alpha_1(x) = \alpha_2(f(x))$ , the substitution is called an identical substitution. The cost of identical vertex or edge substitution is usually assumed to be zero, while the cost of any other edit operation is greater than zero.

<sup>4</sup>It can be considered that all the zero-cells rank first and the rest cells rank from  $\text{null}_i + 1$ .

<sup>5</sup>For weighted graphs, edge label is the weight of the edge.

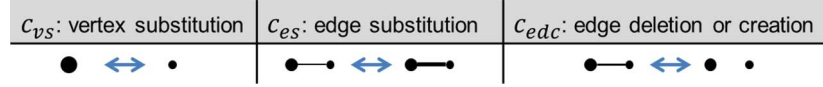


Fig. 6. Three basic graph operators for editing graphs.

**Definition 2:** The cost of an ECGM  $f: \hat{\mathcal{V}}_1 \rightarrow \hat{\mathcal{V}}_2$  from graph  $g_1 = (\mathcal{V}_1, \alpha_1, \beta_1)$  to  $g_2 = (\mathcal{V}_2, \alpha_2, \beta_2)$  is given by

$$\gamma(f, g_1, g_2) = \sum_{x \in \mathcal{V}_1 - \hat{\mathcal{V}}_1} c_{vd}(x) + \sum_{x \in \mathcal{V}_2 - \hat{\mathcal{V}}_2} c_{vi}(x) + \sum_{x \in \hat{\mathcal{V}}_1} c_{vs}(x) + \sum_{e \in \hat{\mathcal{E}}_1} c_{es}(e) \quad (3)$$

where  $c_{vd}(x)$  is the cost of deleting a vertex  $x \in \mathcal{V}_1 - \hat{\mathcal{V}}_1$  from  $g_1$ ,  $c_{vi}(x)$  is the cost of inserting a vertex  $x \in \mathcal{V}_2 - \hat{\mathcal{V}}_2$  in  $g_2$ ,  $c_{vs}(x)$  is the cost of substituting a vertex  $x \in \hat{\mathcal{V}}_1$  by  $f(x) \in \mathcal{V}_2$ , and  $c_{es}(e)$  is the cost of substituting an edge  $e = (x, y) \in \hat{\mathcal{V}}_1 \times \hat{\mathcal{V}}_1$  by  $e' = (f(x), f(y)) \in \hat{\mathcal{V}}_2 \times \hat{\mathcal{V}}_2$ .

**Definition 3:** Let  $f$  be an ECGM from  $g_1$  to  $g_2$ ,  $\mathcal{C}$  a cost function. We call  $f$  an optimal ECGM under  $\mathcal{C}$  if there is no other ECGM  $f'$  from  $g_1$  to  $g_2$  with  $\gamma_{\mathcal{C}}(f', g_1, g_2) < \gamma_{\mathcal{C}}(f, g_1, g_2)$ .

In our cases, if we set the number of face track clusters as the same with the number of character names, the name and face affinity graph have the same number of vertexes. Therefore, there is no need to search for subgraph isomorphisms in scheme 1. We have  $|\hat{\mathcal{V}}_1| = |\mathcal{V}_1| = |\hat{\mathcal{V}}_2| = |\mathcal{V}_2|$ . Also, as no vertex deletion or insertion operation is involved, we can directly assign  $c_{vd} = c_{vi} = \infty$ . According to the investigation on noises, we introduce  $c_{edc}(e)$  for the cost of destroying an edge  $e \in \hat{\mathcal{V}}_1 \times \hat{\mathcal{V}}_1$  or creating an edge  $e \in \hat{\mathcal{V}}_2 \times \hat{\mathcal{V}}_2$ . The edit operation of destroying an edge means certain cell in the name ordinal affinity matrix is nonzero while the corresponding cell in the face ordinal affinity matrix is zero. The edit operation of creating an edge means the opposite. Fig. 6 shows the three basic graph operations defined in this paper. We define the cost of an ECGM in our name/face ordinal affinity graph matching application as

$$\begin{aligned} \gamma(f, g_1, g_2) &= \sum_{x \in \mathcal{V}_1 - \hat{\mathcal{V}}_1} c_{vd}(x) + \sum_{x \in \mathcal{V}_2 - \hat{\mathcal{V}}_2} c_{vi}(x) \\ &+ \sum_{x \in \hat{\mathcal{V}}_1} |\alpha_1(x) - \alpha_2(x)| c_{vs}(x) + \sum_{\substack{\beta_1(e) \cdot \beta_2(e) \neq 0 \\ \beta_1(e) \neq \beta_2(e)}} c_{edc}(e) \\ &+ \sum_{e \in \hat{\mathcal{E}}_1} |\beta_1(e) - \beta_2(e)| c_{es}(e) \end{aligned} \quad (4)$$

where  $|\alpha_1(x) - \alpha_2(x)|$  and  $|\beta_1(e) - \beta_2(e)|$  measure the degree of vertex substitution and edge substitution, respectively.

According to the likelihood of graph distortions during the graph construction process, we assign different costs to the edit operation of vertex substitution, edge substitution and edge creation/destruction. The cost function  $\mathcal{C}$  is designed as

$$\mathcal{C} = (c_{vd}, c_{vi}, c_{vs}, c_{es}, c_{edc}) = (\infty, \infty, \lambda_1, 1, \lambda_2) \quad (5)$$

where  $\lambda_1$  and  $\lambda_2$  embody the likelihood of different graph distortions. Without prior knowledge, we perform experiments on a training set with various value of  $\lambda_1$  and  $\lambda_2$  and select those which maximize the average matching accuracy. Recalling the

example in Fig. 5, apparently no vertex deletion or insertion operation is involved. Also no vertex substitution or edge substitution operations happen. There involve two edge insertion operations (edge  $\{\text{Face3}, \text{Face5}\}$ ,  $\{\text{Face5}, \text{Face3}\}$ ) and one edge substitution operation. The cost of this ECGM under our designed cost function  $\mathcal{C}$  is:  $\gamma_{\mathcal{C}}(f, \hat{R}^{\text{face}}, \hat{R}^{\text{name}}) = 2\lambda_2 + \lambda_1$ .

Consider  $N$  face clusters and character names, the number of possible states for the solution space is the permutation of  $N$ , i.e.,  $N! = N \times (N-1) \times \dots \times 2 \times 1$ . A general algorithm to obtain the optimal ECGM is based on the  $A^*$  method [27]. By applying  $A^*$ , we are able to find the best matching by exploring only the most promising avenues, which guarantees a global optimal.

### III. SCHEME 2: FACE-NAME GRAPH MATCHING WITHOUT NUMBER OF CLUSTER SPECIFIED

Scheme 2 requires no specification for the face cluster number. Standard affinity propagation [29] is utilized for face tracks clustering. The similarity input  $s(i, k)$  is set as the Earth Mover's Distance (EMD) [30] between face tracks, which is same as introduced in [2]. All face tracks are equally suitable as exemplars and the preferences  $s(k, k)$  are set as the median of the input similarities. There are two kind of messages, "availability" and "responsibility," changed between face tracks. With "availability"  $a(i, k)$  initialized to be zero, the "responsibilities"  $r(i, k)$  are computed and updated using the rule

$$r(i, k) \leftarrow s(i, k) - \max_{k', s.t. k' \neq k} \{a(i, k') + s(i, k')\}. \quad (6)$$

While,  $a(i, k)$  is updated using the rule

$$a(i, k) \leftarrow \min \left\{ 0, r(k, k) + \sum_{i', s.t. i' \notin i, k} \max(0, r(i', k)) \right\}. \quad (7)$$

The message-passing procedure converges when the local decisions remain constant for certain number of iterations. In our case, high cluster purity with large number of clusters is encouraged. Therefore, we set the number of iteration as 3 in the experiments, to guarantee concise clusters with consistent appearances.

Since no restriction is set on the one-to-one face-name correspondence, the graph matching method is expected to cope with the situations where several face clusters correspond to the same character name. In view of this, a graph partition step is conducted before graph matching. Traditional graph partition aims at dividing a graph into disjoint subgraphs of the same size [31]. In this paper, graph partition is only used to denote the process of dividing original face graphs. We do not set separate metrics for an optimal graph partition. Instead, graph partition is jointly optimized with the graph matching. This simultaneous process is illustrated in Fig. 7. Instead of separately performing graph partition and graph matching, and using the partitioned face graph

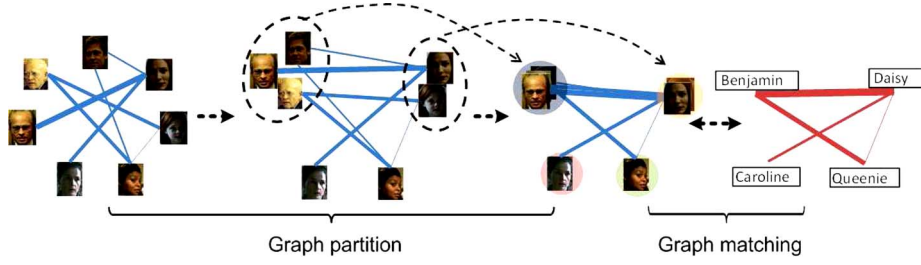


Fig. 7. Simultaneously graph partition and matching for scheme 2.

	Face1	Face2	Face3	Face4	Face5	Face6	Face7
Face1	0.124	0	0	0.125	0	0	0
Face2	0	0.133	0	0	0.031	0.036	0
Face3	0	0	0.024	0.011	0	0.004	0
Face4	0.125	0	0.011	0.175	0	0	0.063
Face5	0	0.031	0	0	0.081	0.002	0
Face6	0	0.036	0.004	0	0.002	0.096	0
Face7	0	0	0	0.063	0	0	0.054

(a)

	Partition 1	Partition 2	Partition 3	Partition 4
Partition 1	0.281	0.167	0.040	0
Partition 2	0.167	0.256	0.002	0.063
Partition 3	0.040	0.002	0.096	0
Partition 4	0	0.063	0	0.054

(b)

	Benja	Daisy	Queen	Carol
Benja	0.241	0.137	0.053	0
Daisy	0.137	0.276	0.001	0.061
Queen	0.053	0.001	0.099	0
Carol	0	0.061	0	0.044

(c)

Fig. 8. Example affinity matrices from the movie “The Curious Case of Benjamin Button.” (a) Original face affinity matrix  $R^{\text{face}}$ . (b) Partitioned face affinity matrix  $\tilde{R}^{\text{face}}(p)$ . (c) Name affinity matrix  $R^{\text{name}}$ .

as input for graph matching, graph partition and graph matching are optimized in a unique framework.

We first define the graph partition  $p$  with respect to the original face graph  $G^{\text{face}}$ . Consider  $N$  character names and  $M$  face track clusters, it divides  $G^{\text{face}}$  into  $N$  disjoint subgraphs

$$p = \{g_1^{\text{face}}, g_2^{\text{face}}, \dots, g_n^{\text{face}}\}. \quad (8)$$

Each subgraph  $g_k^{\text{face}}$  is a sublayer of  $G^{\text{face}}$  with vertex set  $\mathcal{V}_k^{\text{face}}$ , and

$$\begin{aligned} \bigcup_{k=0}^N \mathcal{V}_k^{\text{face}} &= \mathcal{V}^{\text{face}}. \\ \mathcal{V}_i^{\text{face}} &\neq \emptyset, \forall i. \mathcal{V}_i^{\text{face}} \cap \mathcal{V}_j^{\text{face}} = \emptyset, \forall i \neq j \end{aligned} \quad (9)$$

where  $\mathcal{V}^{\text{face}}$  denotes the vertex set of face graph  $G^{\text{face}}$ . In this way, the number of vertexes for each subgraph  $g_k^{\text{face}}$ ,  $|g_k^{\text{face}}| \in \{1, 2, \dots, M - N + 1\}$ ,  $k = 1, 2, \dots, N$ . The vertexes in the same subgraph are considered from the same character and their co-occurrence statistics are integrated. The partitioned face graph has the same vertex number with the name graph. The partitioned face affinity matrix by  $p$ ,  $\tilde{R}^{\text{face}}(p)$  is calculated as

$$\begin{aligned} R_{ii}^{\text{face}}(p) &= \sum_{m \in \mathcal{V}_i^{\text{face}}} r_{mm}^{\text{face}}. \\ R_{ij}^{\text{face}}(p) &= \sum_{\substack{m \in \mathcal{V}_i^{\text{face}}, n \in \mathcal{V}_j^{\text{face}} \\ m \neq n}} r_{mn}^{\text{face}}, \quad i \neq j. \end{aligned} \quad (10)$$

The affinity matrices in Fig. 8 are from the movie “The Curious Case of Benjamin Button”. Fig. 8(b) demonstrates the partitioned face graph by  $p = \{(\text{Face1}, \text{Face2}, \text{Face3}), (\text{Face4}, \text{Face5}), (\text{Face6}), (\text{Face7})\}$  from the original face graph of Fig. 8(a). The partitioned face affinity matrix is then transformed to the corresponding ordinal affinity matrix  $\tilde{R}^{\text{face}}(p)$  according to Section III. Combining

with the ECGM based cost representation, the optimal solution for graph partition and graph matching  $\Theta^* = (p^*, f^*)$  under cost function  $\mathcal{C}$  is obtained by

$$\Theta^* = \arg \min_{p, f} \gamma_{\mathcal{C}}(f, \tilde{R}^{\text{face}}(p), \tilde{R}^{\text{name}}). \quad (11)$$

Consider  $N$  character names and  $M$  face track clusters, there are  $P_M^N N^{M-N}$  possible partitions, where  $P_M^N = M \times (M - 1) \times \dots \times (M - N + 1)$  denotes the  $N$ -permutations of  $M$ . Therefore, the solution space for the joint graph partition and graph matching has  $P_M^N N^{M-N} \cdot N!$  possible states. A simple preprocess is used to filter the candidate partitions. Since there is a very small chance that different faces of the same character appear at the same time, the face clusters having large affinities in the original face matrix are unlikely from the same character. Therefore, we add the following constraint to graph partition:

$$\text{If } r_{ij}^{\text{face}} > T_p, (v_i, v_j) \notin \mathcal{V}_k \cdot i \neq j, k = 1, 2, \dots, N. \quad (12)$$

where  $T_p$  is the threshold to allow certain noises. We set  $T_p = 0.005$  in the experiments. The filtering process will significantly reduce the solution space. For a typical case of 20 face clusters and 10 character names, the original solution space has a huge  $O(10^{18})$  possible states. After filtering, the solution space is reduced to about  $O(10^{12})$ .

#### IV. SENSITIVITY ANALYSIS

Due to the pose, expression, illumination variation as well as occlusion and low resolution problem, inevitable noise is generated during the process of face detection, face tracking and face tracks clustering, which means the derived face graph does not precisely match the name graph. In this sense, the face graph can be seen as a noisy version of the name graph. Therefore, the sensitivity of the graph matching algorithms to the mixed

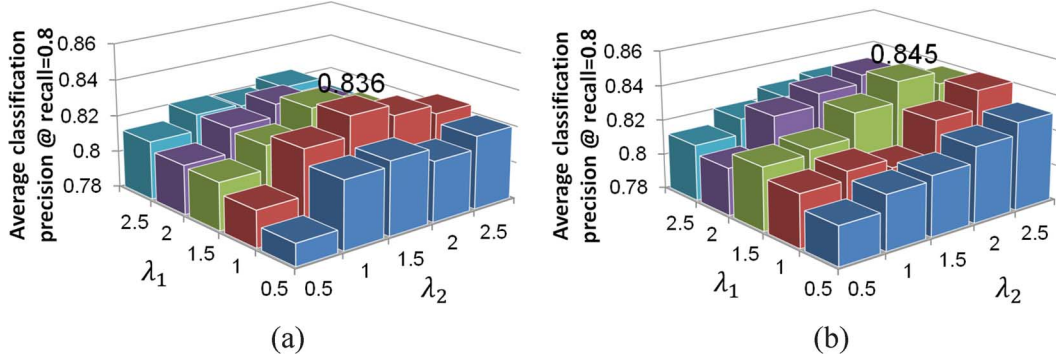


Fig. 9. Average face track classification precision as  $\lambda_1$  and  $\lambda_2$  change. (a) For scheme 1,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$  performs best, with 83.6% precision (b) For scheme 2,  $\lambda_1 = 1.5$ ,  $\lambda_2 = 2$  performs best, with 84.5% precision.

noise is an important metric to evaluate the performance of the method and a sensitivity analysis step has become a must. For movie character identification, sensitivity analysis offers valid tools for characterizing the robustness of the algorithms to the noises from subtitle extraction, speaker detection, face detection and tracking.

In this section, we first introduce two types of simulated noises, coverage noise and intensity noise. The sensitivity score for measuring ordinal graph demotion is defined to evaluate the robustness of the proposed ordinal affinity graph to noises. The sensitivity analysis is further discussed in the experiment section.

#### A. Coverage Noise and Intensity Noise

According to the noise investigation, vertex substitution, edge substitution and edge destruction/creation are involved in the graph construction process for character identification. According to that, we introduce two types of noises, coverage noise and intensity noise for simulation.

Preserving graph connectivity structure is very important in graph matching [32]. We use the graph edit operations of edges creation and destruction to simulate the changes to the topology of the graph. We call it coverage noise. The creation/destruction probability  $\nu_C$  for each existing or potential edge denotes the coverage noise level. Intensity noise corresponds to changes in the weights of the edges. It has involvement with the quantitative variation of the edges, but with no affection to the graph structure. A random value distributed uniformly on the  $[-\nu_I, \nu_I]$  range denotes the intensity noise level. As the affinity values are normalized, the intensity change is limited so as to never take the resulting weight outside of the  $[0, 1]$  interval. Based on the defined noises, we perform sensitivity analysis in the experiments by investigating the performance of name-face matching.

#### B. Ordinal Graph Sensitivity Score

To evaluate the sensitivity of the proposed ordinal affinity graph, the sensitivity score for ordinal graph demotion is specially defined.

The sensitivity score function should be consistent with the likelihood of the graph distortion i.e., noises. Therefore, we de-

fine the ordinal graph sensitivity score  $\mu$  in accordance to our definition of the cost function for ECGM

$$\mu = \sum_{x \in \hat{\mathcal{V}}_1} \lambda_1 |\alpha_1(x) - \alpha_2(x)| + \sum_{e \in \hat{\mathcal{E}}_1} |\beta_1(e) - \beta_2(e)| + \sum_{\substack{\beta_1(e) \cdot \beta_2(e) \neq 0 \\ \beta_1(e) \neq \beta_2(e)}} \lambda_2 \quad (13)$$

where  $\lambda_1$  and  $\lambda_2$  are the same parameters with (5), and  $g_1$  means the ordinal graph before demoted by the noises and  $g_2$  is the corresponding demoted ordinal graph.

## V. EXPERIMENTS

#### A. Dataset

Our data set consists of 15 feature-length movies: “12 Angry Men,” “Inception,” “Se7en,” “Forrest Gump,” “Mission: Impossible,” “Léon: The Professional,” “The Shawshank Redemption,” “Revolutionary Road,” “Legally Blond,” “Devil Wears Prada,” “You’ve Got Mail,” “Sleepless in Seattle,” “Pretty Woman,” “Notting Hill,” and “The Curious Case of Benjamin Button” (corresponding to the ID of F1–F15). The genres range from romance, comedy, to crime and thriller. It is split into two parts: movies of F1–F3 as the training part and the rest 12 movies of F4–F15 as the testing part. The training set has 1 327 face tracks, and the testing set has 5 012 tracks.

#### B. Experimental Results

1) *Cost Function for ECGM*: The costs for different graph edit operations are designed by automatic inference based on the training set. Parameters  $\lambda_1$  and  $\lambda_2$  in (5) and (13) embody the likelihood of different graph distortions. We perform experiments with various values of  $\lambda_1$  and  $\lambda_2$ .

Similar to [18] and [2], we set a threshold  $Th_{score}$  to discard noise before matching. The face tracks with function scores lower than  $Th_{score}$  are refused to classify to any of the clusters and will be left unlabeled. *Recall* here means the proportion of tracks which are assigned a name, and *precision* is the proportion of correctly labeled tracks [18]. Their calculation is given



TABLE I  
FACE TRACK DETECTION ACCURACY

Clip	#Face track	#Track detected	Accuracy
1	372	354	95.2%
2	468	431	92.1%
3	515	472	91.7%

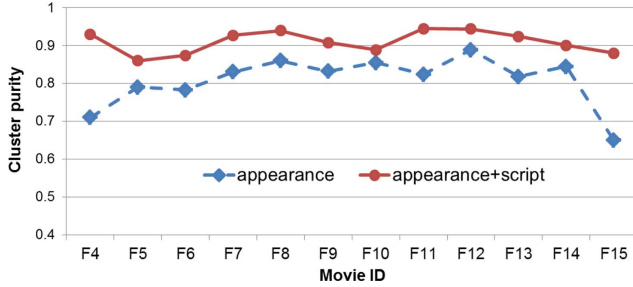


Fig. 10. Cluster purity for the different clustering scheme.

as follows:

$$\text{precision} = \frac{|\text{facetracks correctly classified}|}{|\text{facetracks classified}|} \quad (14)$$

$$\text{recall} = \frac{|\text{facetracks classified}|}{|\text{total facetracks}|}. \quad (15)$$

We use face track classification *precision* to tune the parameters for the cost function.

The result of average face track classification  $\text{precision@recall} = 0.8$  with respect to  $\lambda_1$  and  $\lambda_2$  is shown in Fig. 9. For scheme 1,  $\lambda_1 = 1$  and  $\lambda_2 = 1.5$  perform best, with 83.6% average precision. For scheme 2,  $\lambda_1 = 1.5$  and  $\lambda_2 = 2$  perform best, with 84.5% average precision. Beyond or below the value, the results deteriorate: the derived parameter values embody the difference of likelihood of vertex substitution and edge creation/destruction operations from edge substitution. This means that coverage noise (vertex substitution and edge creation/destruction) are less likely to occur, and the graph topology is relatively stable during the construction process. We fix  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$  for scheme 1 and  $\lambda_1 = 1.5$ ,  $\lambda_2 = 2$  for scheme 2 in the following experiments.

2) *Face Track Detection*: We utilized a multiview face tracker [33] to detect and track faces on each frame. One typical movie contains up to 100 000 faces detected on all the frames, which are derived from a few thousand tracks. Each face track contains about dozens of faces and the length of one face track ranges from several seconds to dozens of seconds. Since the face track detection generally follows Zhang's work [2], we conducted a simple evaluation experiment by randomly selecting three 30-min clips from three movies F4, F5, F6 to accessing the face track detection accuracy. The statistics of the performance are shown in Table I, where “#Face track” and “#Track detected” are the numbers of ground-truth tracks and detected tracks.

3) *Face Track Clustering*: In scheme 1, we follow the same face track clustering steps as in [2]. In scheme 2, the clustering process is actually divided into two parts: face track clustering

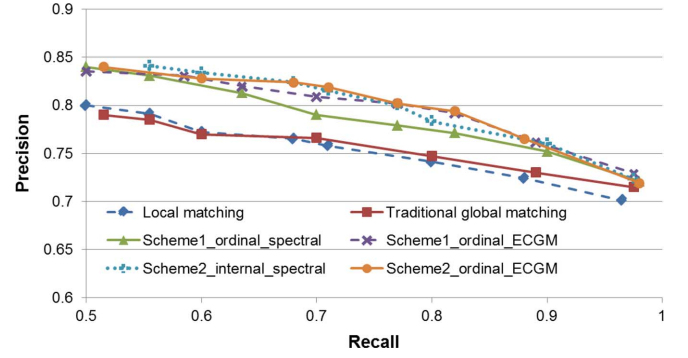


Fig. 11. Precision/recall curves of face track classification.

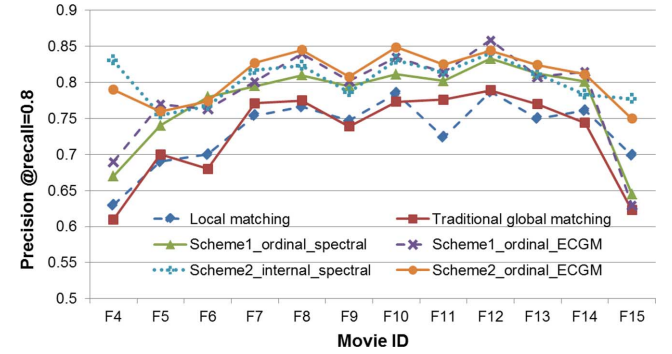


Fig. 12. Face track classification  $\text{precision@recall} = 0.8$ .

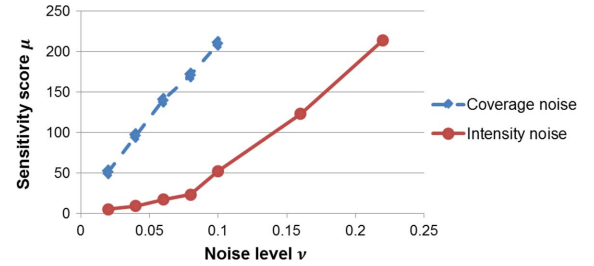


Fig. 13. Sensitivity score  $\mu$  v.s. noise level for coverage noise and intensity noise.

by appearance and graph partition by utilizing script. Cluster purity is used to evaluate the performance of face clustering:

$$\text{purity} = \frac{1}{N} \sum_{i=1}^N \frac{\max_j |C_i \cap \text{Name}_j|}{|C_i|} \quad (16)$$

where  $C_i$  is the set of face tracks in the  $i$ th cluster and  $\text{Name}_j$  is the set of face tracks with the  $j$ th label (character name). Since high purity is easy to achieve when the number of clusters is large, therefore, for comparison convenience, we define the face track clustering results in scheme 2 as the optimized partition results, i.e., the face tracks in the same subgraph constitute one cluster. In this way, the number of clusters for scheme 1 is the same with that for scheme 2.

We denote the face track clustering only by appearance (as used in [2] and scheme 1) as *appearance*, and the one combining appearance and script information (as used in scheme 2) as *appearance+script*. The result is shown in Fig. 10. It is seen that *appearance+script*-based clustering obtains a consistent higher

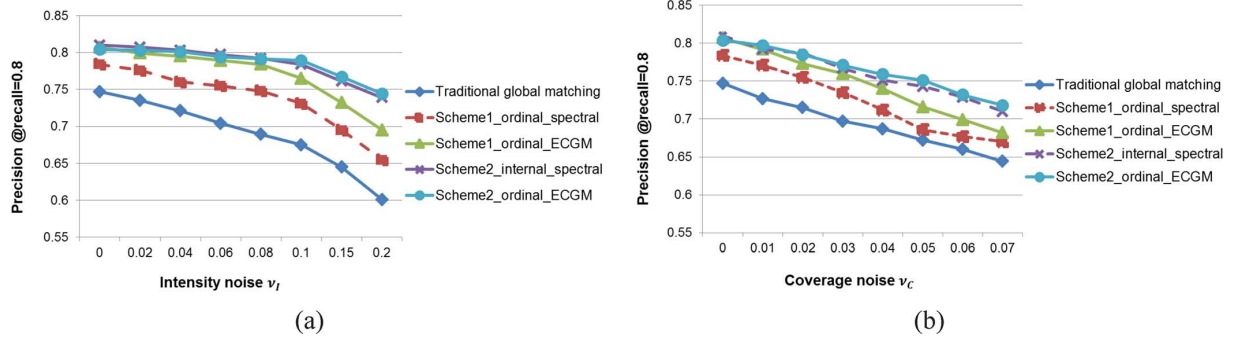


Fig. 14. Precision@recall = 0.8 versus simulated noise level. (a) Intensity noise. (b) coverage noise.

purity than the *appearance*-based. The improvement is from 5% to as high as 30%. The two-step clustering method, which utilizes both the appearance information and the script cues, makes the first step cluster focus on appearance clustering and leave the face-name correspondence issue to the second step. The *appearance*-based method tends to fail when the movie tells a lifetime story. For example, in the movie F4 and F15, the average cluster purity from *appearance*-based clustering is only 71% and 65%. Their stories go along the whole growth of the hero and heroine, where the appearance of characters change dramatically. Instead of helping faces from the same character group together, enforcing the cluster number the same with the character number will deteriorate the clustering process and generate mixed clusters. For the *appearance+script*-based clustering method, the faces of the same character are expected to be grouped into several clusters, with each cluster concerning one period. Using the script cues, the faces of different periods are merged together by graph matching optimization.

4) *Face Track Classification*: A comparison with the existing local matching and global matching approaches was carried out. The approach proposed in [18] and [2] were evaluated on the same test set, which are denoted as *Local matching* and *Traditional global matching* respectively. In addition, we also compared the performances of the proposed approach with different settings: 1) scheme 1 with ordinal graph representation and spectral graph matching (*Scheme1\_ordinal\_spectral*) 2) scheme 1 with ordinal graph representation and ECGM-based graph matching (*Scheme1\_ordinal\_ECGM*) 3) scheme 2 with original internal graph representation and spectral graph matching (*Scheme2\_internal\_spectral*) and 4) scheme 2 with ordinal graph representation and ECGM-based graph matching (*Scheme2\_ordinal\_ECGM*).

Fig. 11 illustrates the face track classification *precision/recall* curve for different character identification algorithms. The detailed *precision@recall=0.8* for the test movies is shown in Fig. 12. We have four observations: 1) The proposed scheme 1 and scheme 2 perform better than the local matching and traditional global matching methods. At the low levels of recall, the proposed methods show convinced improvement. For movies involving with severe variation of face pose and illumination, e.g., action and thriller movie of F5 and F6, the improvement is remarkable. This demonstrates the robustness of the proposed methods to the noises. 2) Generally, the results of scheme 1 is comparable to that of scheme 2. Scheme 2 has a little higher *precision* when  $\text{recall} \in [0.7, 0.8]$ . However, for the individual

movies, scheme 2 shows significant advantages over scheme 1 on movies with complex character changes, e.g., movie F4 and F15. 3) The improvement of *Scheme1\_ordinal\_spectral* over *Traditional global matching* shows the ability of ordinal graph representation for tolerating noises. ECGM provides a way of optimizing cost functions, which leads to the better performance of *Scheme1\_ordinal\_ECGM*. 4) The role of ordinal graph representation and ECGM is not so significant for scheme 2. The improvement of *Scheme2\_ordinal\_ECGM* over *Scheme2\_internal\_spectral* is very small. The reason is that the mechanism of jointly optimizing graph partition and graph matching can also handle certain noises. Direct combination attributes to limited improvement.

5) *Sensitivity Analysis*: Random coverage and intensity noise of different noise levels are generated for the sensitivity analysis. We note that for scheme 2, the simulated noises are added on the original affinity graphs. We first present the curve of ordinal graph sensitivity score change in Fig. 13. It demonstrates how sensitive the average precision of name-face matching is with respect to different coverage noise levels and intensity noise levels, respectively. It is shown that the ordinal graph is more stable towards the intensity noise. When intensity noise  $\nu_C \leq 0.08$ , the sensitivity score remains stable.

Fig. 14 shows the curve of face track classification *precision@recall = 0.8* versus the simulated noises. From Fig. 14(a), we can see that the proposed scheme 1 and scheme 2 basically remain stable when the intensity noise  $\nu_I \leq 0.08$ , while *Traditional global matching* tends to decline even when  $\nu_I = 0.02$ . From Fig. 14(b), coverage noise deteriorates the classification precision of the proposed schemes as well as *Traditional global matching*. However, scheme 2 has a better tolerance to the coverage noises than scheme 1. This is because the simulated noises are added before the graph partition step, optimizing the graph partition helps reduce the sensitivity to noises. We reach the conclusion that the proposed methods are more robust to the intensity noises than to the coverage noises, i.e., ordinal representation and simultaneous graph partition with graph matching have the ability to tolerate the random variations to the values of weighted edges and manage to match graphs correctly as long as the topological structure is preserved. This finding is of great importance. According to our observation and experimental results, though the weights of face affinity relations are imprecise, basically the generated name and face affinity graph have the same topology. On one hand, this serves as one of the validations that we treat zero-cell

different from the nonzero-cell in constructing ordinal graph. On the other hand, the design of robust character identification method needs focusing more on handling the intensity noises.

## VI. CONCLUSION

We have shown that the proposed two schemes are useful to improve results for clustering and identification of the face tracks extracted from uncontrolled movie videos. From the sensitivity analysis, we have also shown that to some degree, such schemes have better robustness to the noises in constructing affinity graphs than the traditional methods. A third conclusion is a principle for developing robust character identification method: intensity alike noises must be emphasized more than the coverage alike noises.

In the future, we will extend our work to investigate the optimal functions for different movie genres. Another goal of future work is to exploit more character relationships, e.g., the sequential statistics for the speakers, to build affinity graphs and improve the robustness.

## REFERENCES

- [1] J. Sang, C. Liang, C. Xu, and J. Cheng, "Robust movie character identification and the sensitivity analysis," in *Proc. ICME*, 2011, pp. 1–6.
- [2] Y. Zhang, C. Xu, H. Lu, and Y. Huang, "Character identification in feature-length films using global face-name matching," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1276–1288, Nov. 2009.
- [3] M. Everingham, J. Sivic, and A. Zisserman, "Taking the bite out of automated naming of characters in TV video," *J. Image Vis. Comput.*, pp. 545–559, 2009.
- [4] C. Liang, C. Xu, J. Cheng, and H. Lu, "Tvparsr: An automatic tv video parsing method," in *Proc. Comput. Vis. Pattern Recognit.*, 2011, pp. 3377–3384.
- [5] J. Sang and C. Xu, "Character-based movie summarization," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 855–858.
- [6] R. Hong, M. Wang, M. Xu, S. Yan, and T.-S. Chua, "Dynamic captioning: Video accessibility enhancement for hearing impairment," *ACM Trans. Multimedia*, pp. 421–430, 2010.
- [7] T. Cour, B. Sapp, C. Jordan, and B. Taskar, "Learning from ambiguously labeled images," in *Proc. Comput. Vis. Pattern Recognit.*, 2009, pp. 919–926.
- [8] J. Stallkamp, H. K. Ekenel, and R. Stiefelhagen, "Video-based face recognition on real-world data," in *Proc. Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
- [9] S. Satoh and T. Kanade, "Name-it: Association of face and name in video," in *Proc. Comput. Vis. Pattern Recognit.*, 1997, pp. 368–373.
- [10] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y. W. Teh, E. G. Learned-Miller, and D. A. Forsyth, "Names and faces in the news," in *Proc. Comput. Vis. Pattern Recognit.*, 2004, pp. 848–854.
- [11] J. Yang and A. Hauptmann, "Multiple instance learning for labeling faces in broadcasting news video," in *Proc. ACM Int. Conf. Multimedia*, 2005, pp. 31–40.
- [12] A. W. Fitzgibbon and A. Zisserman, "On affine invariant clustering and automatic cast listing in movies," in *Proc. ECCV*, 2002, pp. 304–320.
- [13] O. Arandjelovic and R. Cipolla, "Automatic cast listing in feature-length films with anisotropic manifold space," in *Proc. Comput. Vis. Pattern Recognit.*, 2006, pp. 1513–1520.
- [14] D. Ramanan, S. Baker, and S. Kakade, "Leveraging archival video for building face datasets," in *Proc. Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
- [15] M. Everingham and A. Zisserman, "Identifying individuals in video by combining "generative" and discriminative head models," in *Proc. Int. Conf. Comput. Vis.*, 2005, pp. 1103–1110.
- [16] R. G. Cinbis, J. Verbeek, and C. Schmid, "Unsupervised metric learning for face identification in TV video," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 1559–1566.
- [17] M. Xu, X. Yuan, J. Shen, and S. Yan, "Cast2face: Character identification in movie with actor-character correspondence," *ACM Multimedia*, pp. 831–834, 2010.
- [18] M. Everingham, J. Sivic, and A. Zisserman, "Hello! my name is... buffy automatic naming of characters in tv video," in *Proc. BMVC*, 2006, pp. 889–908.

- [19] J. Sivic, M. Everingham, and A. Zisserman, "Who are you?—Learning person specific classifiers from video," in *Proc. Comput. Vis. Pattern Recognit.*, 2009, pp. 1145–1152.
- [20] T. Cour, C. Jordan, E. Mitsakaki, and B. Taskar, "Movie/script: Alignment and parsing of video and text transcription," in *Proc. ECCV*, 2008, pp. 158–171.
- [21] T. Cour, B. Sapp, A. Nagle, and B. Taskar, "Talking pictures: Temporal grouping and dialog-supervised person recognition," in *Proc. Comput. Vis. Pattern Recognit.*, 2010, pp. 1014–1021.
- [22] Y. Zhang, C. Xu, J. Cheng, and H. Lu, "Naming faces in films using hypergraph matching," in *Proc. ICME*, 2009, pp. 278–281.
- [23] A. Saltelli, M. Ratto, S. Tarantola, and F. Campolongo, "Sensitivity analysis for chemical models," *Chem. Rev.*, vol. 105, no. 7, pp. 2811–2828, 2005.
- [24] E. Bini, M. D. Natale, and G. Buttazzo, "Sensitivity analysis for fixed-priority real-time systems," *Real-Time Systems*, vol. 39, no. 1, pp. 5–30, 2008.
- [25] R. E. Hanneman, *Introduction to Social Network Methods*. Riverside, CA: Univ. California, 2000, Online Textbook Supporting Sociology 157.
- [26] E. Bengoetxea, "Inexact graph matching using estimation of distribution algorithms," Ph.D. dissertation, Ecole Nationale Supérieure des Télécommunications, Paris, France, 2003.
- [27] A. Sanfeliu and K. Fu, "A distance measure between attributed relational graphs for pattern recognition," *IEEE Trans. Syst. Man Cybern.*, vol. 13, no. 3, 1983.
- [28] H. Bunke, "On a relation between graph edit distance and maximum common subgraph," *Pattern Recognit. Lett.*, vol. 18, pp. 689–694, 1997.
- [29] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–977, 2007.
- [30] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proc. Int. Conf. Comput. Vis.*, 1998, pp. 59–66.
- [31] L. Lin, X. Liu, and S. C. Zhu, "Layered graph matching with composite cluster sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1426–1442, Aug. 2010.
- [32] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Comput. Vis. Image Understand.*, vol. 89, no. 2–3, pp. 114–141, 2003.
- [33] 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.



**Jitao Sang** received the B.E. degree from the South-East University, Nanjing, China, in 2007 and is currently pursuing the Ph.D. degree at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing.

In 2010 and 2011, he was an intern student in the China-Singapore Institute of Digital Media (CSIDM) and Microsoft Research Asia (MSRA), respectively. His current research interests include multimedia content analysis, social media mining, computer vision and pattern recognition.



**Changsheng Xu** (M'97-SM'99) is Professor in National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences and Executive Director of China-Singapore Institute of Digital Media. His research interests include multimedia content analysis/indexing/retrieval, pattern recognition and computer vision. He has held 30 granted/pending patents and published over 200 refereed research papers in these areas.

Dr. Xu is an Associate Editor of ACM Transactions on Multimedia Computing, Communications and Applications and ACM/Springer Multimedia Systems Journal. He served as Program Chair of ACM Multimedia 2009. He has served as associate editor, guest editor, general chair, program chair, area/track chair, special session organizer, session chair and TPC member for over 20 IEEE and ACM prestigious multimedia journals, conferences and workshops.