Recognisation & Identification of Automatic Face-Name Detection by using Spectral Graph Matching Algorithm

Kanike Vijay Kumar[1] P. Kishor Kumar[2]
[1,2]Assistant Professor
[1,2]Department of Electronics & Communication Engineering
[1,2]Ravindra College of Engineering for Women, Kurnool, A.P., India

Abstract— In Real Time, 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. 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.

Key words: Character Identification, Graph Edit, Graph Matching, Graph Partition, Sensitivity Analysis

I. Introduction

A. Objective & 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. The annotating characters in the movie and TVs, which is called movie character identification. The objective is to identify the faces of the characters in the video and label them with the corresponding names in the cast. In a movie, characters are the focus centre of interests for the audience. Automatic character identification is essential for semantic movie index and retrieval, scene segmentation, summarization and other applications.

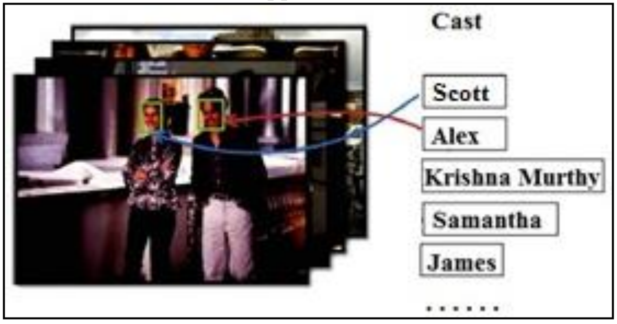


Fig. 1: Examples of Character Identification from Movie "Notting Hill."

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. There are ambiguity problem in establishing the correspondence between names and faces. Ambiguity can arise in partially labelled frames when there are multiple speakers in the same scene2.
2) Face identification in videos is more difficult than that in images. Low resolution, occlusion, non-rigid deformations, large motion, complex background and other uncontrolled conditions make the results of face detection and tracking unreliable.
3) The same character appears quite differently during the movie. There may be huge pose, expression and illumination variation, wearing, clothing, even makeup and hairstyle changes.
4) The determination for the number of identical faces is not trivial. 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.

B. Related Work

The crux of the character identification problem is to exploit the relations between videos and the associated texts in order to label the faces of characters with names. It has similarities to identifying faces in news videos. 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, faces are clustered by appearance and faces of a particular character are expected to be collected in a few pure clusters. An interesting work combining character identification with web image retrieval is proposed. The character names in the cast are used as queries to search face images and constitute gallery set.

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. The combine the film script with the subtitle for local face-name matching. 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. 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.
- 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, the number of clusters is the same as the number of distinct speakers appearing in the script.
- Sensitivity analysis is common in financial applications, risk analysis, signal processing and any area where models are developed.

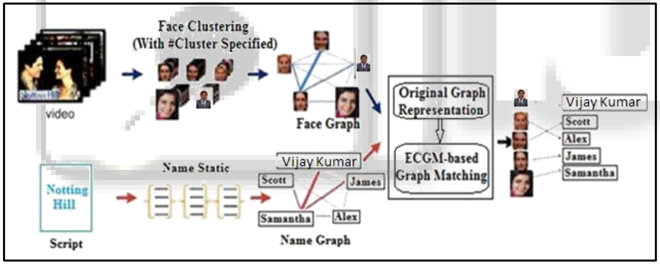


Fig. 2: Framework of Scheme 1: Face-Name Graph Matching with #Cluster Pre-Specified

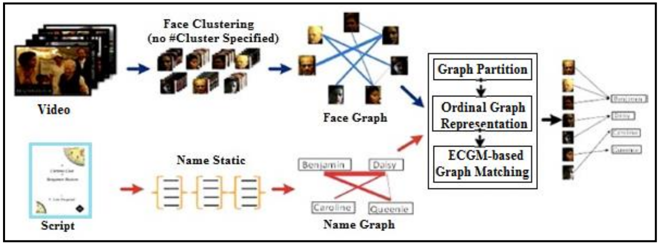


Fig. 3: Framework of Scheme 2: Face-Name Graph Matching without #Cluster Prespecified

For movie character identification, sensitivity analysis offers valid tools for characterizing the robustness to noises for a model. There have been no efforts directed at the sensitivity analysis for movie character identification. Main aim is to fill this gap by introducing two types of simulated noises.

C. Overview of Our Approach

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. While, in scheme 2, no cluster number is required and face tracks are clustered based on their intrinsic data structure.

1) Scheme 1

The proposed framework for scheme 1 is shown in Fig. 2. Co-occurrence of names in script and face clusters in video constitutes the corresponding face graph and name graph. For face and name graph construction, we propose to represent the character co-occurrence in rank ordinal level, which scores the strength of the relationships in a rank order from the weakest to strongest. 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.

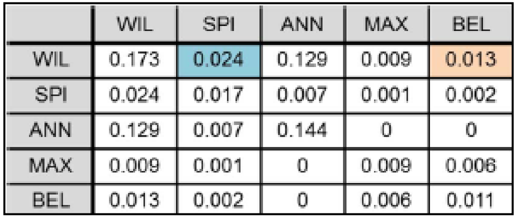
2) Scheme 2

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. It is difficult to group them in a unique cluster.

In scheme 2, we utilize affinity propagation for the face tracks clustering. High cluster purity with large number of clusters is expected. 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.

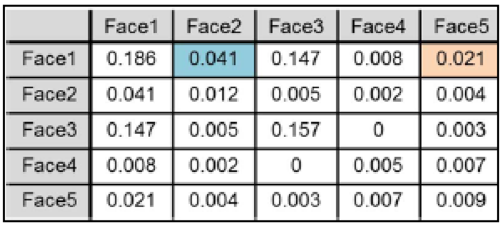
In general, the scheme 2 has two advantages over the scheme 1

1) For scheme 2, no cluster number is required in advance and face tracks are clustered based on their intrinsic data structure.
2) 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.
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.



	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)

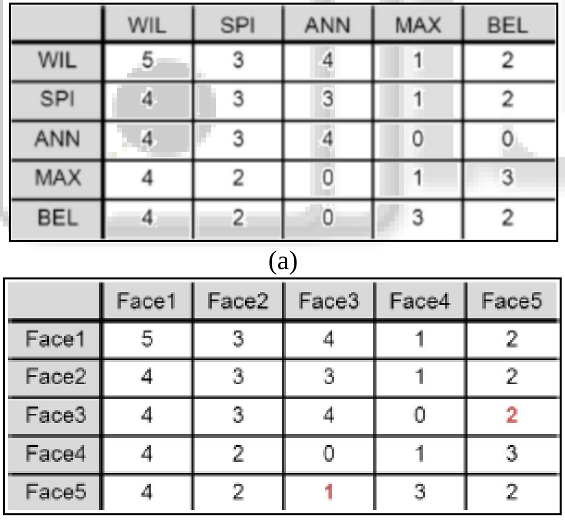
Fig. 4: Example of Affinity Matrices from Movie "Notting Hill":
(a) Name affinity matrix R^{name} (b) Face affinity matrix R^{face}.

II. Scheme 1: Face-Name Graph Matching with Number of Cluster Specified

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*

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 E edge denotes relationships among them.



	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 (b) Face ordinal affinity matrix

Fig. 4 demonstrates the adjacency matrices corresponding to the name and face affinity graphs from the movie "Noting Hill"3. 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 value due to the introduced noises. A spectral graph matching algorithm is applied to find the optimal name-face correspondence.

B. *Ordinal Graph Representation*

The name affinity graph and face affinity graph are built based on the co-occurrence relationship. 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.

We denote the original affinity matrix as R= $\{r_{ij}\}_{NxN}$, 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_{ij} in ascending order, then the corresponding diagonal cells in the rank ordinal affinity matrix

$$\tilde{r}_{ii} = I_{rii} \tag{1}$$

1where I_{rii} is the rank index of original diagonal affinity value r_{ii} . Zero-cell represents that no co-occurrence relationship is specially considered, which a qualitative measure is. 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 ith row is recorded as $null_{i.}$ For the ith row, the corresponding cells $\tilde{r}_{ii}$ in the ith row of ordinal affinity matrix

$$\tilde{r}_{ij} = I_{rij} + null_i \tag{2}$$

Where I_{rij} denotes the order of r_{ij}. The scales reflect variances in degree of intensity, but not necessarily equal differences. 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.

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. 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.

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 is utilized for face tracks clustering. The similarity input $s(i,k)$ is set as the Earth Mover's Distance (EMD) between face tracks. 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

$$r(i,k) \leftarrow s(i,k) - \max_{k^1,s,t,k^1 \neq k} \{a(i,k^1) + s(i,k^1)\} \tag{3}$$

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

$$a(i,k\ ) \leftarrow \min\{0, r(k,k) + \sum_{i^1,s,t,i^1 \notin i,k} \max 0, r(i^1,k)\} \tag{4}$$

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.

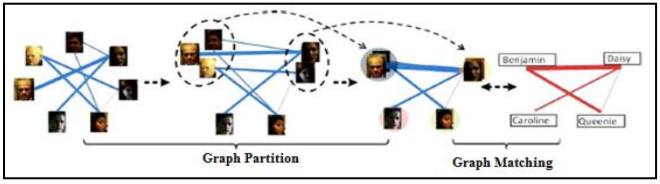
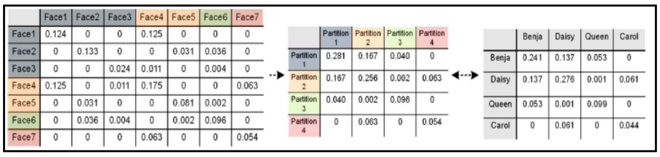


Fig. 6: Simultaneously Graph Partition and Matching for Scheme 2



	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

	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

	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

(a) (b) (c)

Fig. 7: Example Affinity Matrices from the Movie “The Curious Case of Benjamin Button.” (a) Original Face Affinity Matrix (b) Partitioned Face Affinity Matrix (c) Name Affinity Matrix

This simultaneous process is illustrated in Fig. 7. Instead of separately performing graph partition and graph matching, and using the partitioned face graph as input for graph matching, graph partition and graph matching are optimized in a unique framework.

We first define the graph partition with respect to the original face graph G^{face}. Consider N character names and M face track clusters, it divides G^{face} into N disjoint sub-graphs

$$p = \{g_1^{face}, g_2^{face}, \ldots., g_n^{face}\} \tag{5}$$

Each subgraph g_n^{face} is a sublayer of G^{face} with vertex set v_k^{face}, and

$$U_{k=0}^{n} v_k^{face} = v^{face} \tag{6}$$

$$v_i^{face} \neq \phi, \forall_i . v_i^{face} \cap v_j^{face} = \phi, \forall_i \neq j$$

Where v^{face} denotes the vertex set of face graph G^{face}. In this way, the number of vertexes for each subgraph $g_k^{face}, \left|g_k^{face}\right| \in \{1,2,..,M-N+1\}, k = 1,2,..,N$ the partitioned face affinity matrix $p, R^{face}(p)$ by is calculated as

$$R_{ii}^{face}(p) = \sum_{m \in v_i^{face}} r_{mm}^{face}$$

$$R_{ij}^{face}(p) = \sum_{m \in v_i^{face}, n \in v_j^{face} m \neq n} r_{mn}^{face}, i \neq j. \tag{7}$$

The affinity matrices in Fig. 7 are from the movie “The Curious Case of Benjamin Button”. Fig. 7(b) demonstrates the partitioned face graph by $p = \{(Face1, Face2, Face3), (Face4, Face5), (Face6), (Face7)\}$ from the original face graph of Fig. 7(a).

IV. Sensitivity Analysis

Due to the 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. 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.

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.

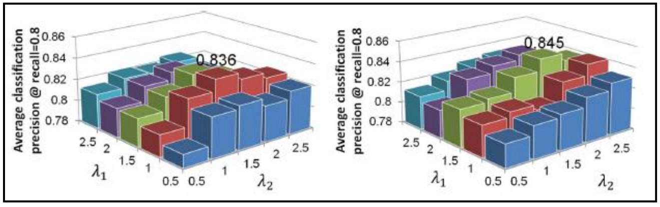


Fig. 8: Average Face Track Classification Precision as and Change. (a) For Scheme 1, Performs Best, with 83.6% Precision (b) For Scheme 2, Performs Best, with 84.5% precision

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 $[-v_I, v_I]$ range denotes the intensity noise level.

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 define the ordinal graph sensitivity score in accordance to our definition of the cost function for ECGM

$$\mu = \sum_{x \in v_1} \lambda_1 \left|\alpha_1(x) - \alpha_2(x)\right| + \sum_{e \in \hat{\varepsilon}_1} \left|\beta_1(e) - \beta_2(e)\right| + \sum_{\beta_1(e).\beta 2(e) \neq 0 \& \beta_1(e) \neq \beta 2(e)} \lambda_2 \tag{8}$$

Where λ_1 and λ_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. 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 λ_1 and λ_2 in equ.(8) embody the likelihood of different graph distortions.

Recall here means the proportion of tracks which are assigned a name, and precision is the proportion of correctly labeled tracks. Their calculation is given as follows:

$$precision = \frac{|\text{facetracks correctly classified}|}{|\text{facetracks classified}|} \quad (9)$$

$$recall = \frac{|\text{facetracks classified}|}{|\text{total classified}|} \quad (10)$$

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

The result of average face track classification $precision@recall = 0.8$ with respect to λ_1 and λ_2 is shown in Fig. 8. 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.

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

Table 1: Face Track Detection Accuracy

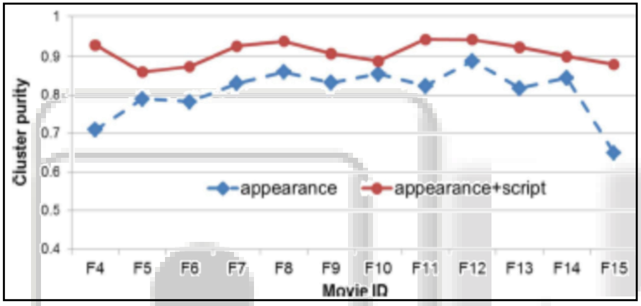


Fig. 9: Cluster purity for the different clustering scheme

Beyond or below the value, the results deteriorate. The graph topology is relatively stable during the construction process.

2) Face Track Detection

We utilized a multi view face tracker to detect and track faces on each frame. 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. In scheme 2, the clustering process is actually divided into two parts: face track clustering by appearance and graph partition by utilizing script. Cluster purity is used to evaluate the performance of face clustering: $purity = \frac{1}{N}\sum_{i=1}^{N}\frac{\max\left|C_i \cap Name_j\right|}{\left|C_i\right|}$

(11) where C_i the set of face is tracks in the ith cluster and $Name_j$ is the set of face tracks with the jth label (character name).

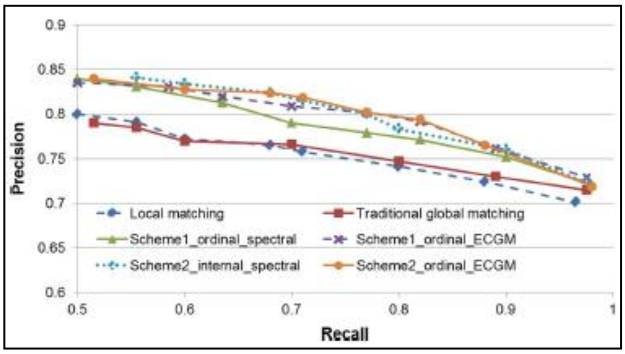


Fig. 10: Precision/Recall curves of Face Track Classification. Since High Purity is Easy to Achieve when the Number of Clusters is Large.

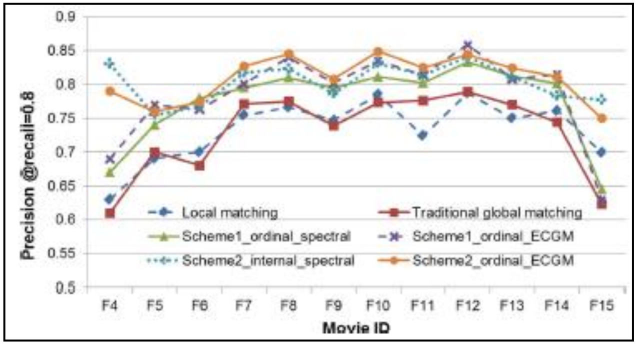


Fig. 11: Face Track Classification

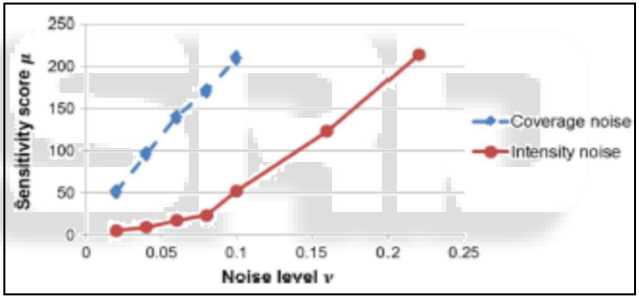


Fig. 12: Sensitivity Score v.s. Noise Level for Coverage Noise and Intensity

Noise

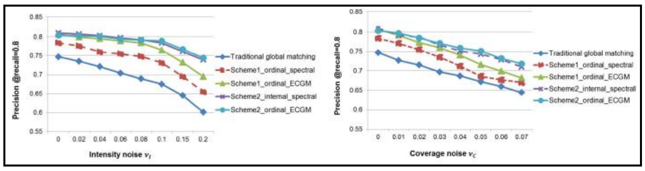


Fig. 13: Precision@recall=0.8 versus Simulated Noise Level. (a) Intensity Noise. (b) Coverage Noise

Face Track Classification: A comparison with the existing local matching and global matching approaches was carried out. The approach proposed is evaluated on the same test set, which are denoted as Local matching and Traditional global matching respectively.

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.

References

[1] C. Liang,C.Xu, J. Cheng, andH.Lu, "Tvparser: An automatic tv video parsing method," in Proc. Comput. Vis. Pattern Recognit., 2011, pp. 3377–3384.

[2] Y. Zhang, C. Xu, H. Lu, and Y. Huang, "Character identification in feature-length films using global face-name matching," IEEE Trans. ultimedia, vol. 11, no. 7, pp. 1276–1288, Nov. 2009.

[3] J. Sang, C. Liang, C. Xu, and J. Cheng, "Robust movie character identification and the sensitivity analysis," in Proc. ICME, 2011, pp. 1–6.

[4] J. Sang and C. Xu, "Character-based movie summarization," in Proc. ACM Int. Conf. Multimedia, 2010, pp. 855–858.

[5] 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.

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