AUTOMATIC SOCIAL NETWORK CONSTRUCTION FROM MOVIES USING FILM-EDITING CUES

Mei-Chen Yeh, Ming-Chi Tseng, Wen-Po Wu

Department of Computer Science and Information Engineering
National Taiwan Normal University, Taipei, Taiwan
myeh@csie.ntnu.edu.tw

ABSTRACT

In this paper, we investigate the problem of automatically constructing characters' social network from movies. Unlike existing approaches that use co-appearance information to measure the relationship between two characters, we argue that a method that describes the characters' interaction, rather than the co-appearance, makes more sense. We propose a new scheme that quantifies the interaction of characters by the use of film-editing cues, based on which we construct the characters' social network. Experiments on real-world data validate the effectiveness of the proposed method. In addition, we show an application of discovering characters' social clusters enabled by the automatically constructed social network.

Index Terms— Social network, face clustering, social cluster discovery

1. INTRODUCTION

Social network techniques have gained increasing attentions in movie content analysis as they provide an alternative approach to audiovisual features for organizing explosive amounts of movie data. A social network is a collection of relationships that demonstrate how a group of people are socially connected to one another [12]. A weighted undirected graph is usually used to represent the social context information where vertices denote the people and edges indicate the social closeness of two individuals. The analyses of social networks have shown some success in discovering hidden structures that cannot be directly perceived by analyzing low-level features, and enabled many interesting applications such as video understanding [13], story segmentation [9][10], face clustering [11], and photo annotation [8][6].

Given a movie, roles' social network [1] should be constructed with minimal human intervention to truly aid in the movie content organization or retrieval process. Modeling characters' interrelationship in a movie is one of the main challenges in building the social network by a computational approach. Most existing methods utilize the co-appearance to quantify characters' interrelationship—the social closeness of two characters is measured by the number of scenes where both characters are present [10][12]. Precise scene boundaries are required in those approaches. The social network is semi-automatically constructed by first applying a scene detection method to obtain initial scene boundaries. Then, users are required to get involved in the process in order to correct errors caused from the scene detection method.

In this paper, we argue that the interrelationship of characters may be more appropriately represented by the interaction compared to the co-appearance. The reasons are two-fold. First, two characters can still interact even if they do not physically appear in the same space at the same time. For example, a person may communicate with others through a phone. Second, the fact that two people are present in a same scene does not imply that they would interact with each other. For example, the camera may capture a person who appears in background and he/she may not interact with other characters throughout the scene. In this paper, we propose an alternative method for qualifying characters' interrelationship by their interactions, and such a method does not require scene boundary information. As will be shown later in Section 3, the idea can be easily implemented by exploiting the rich structured information in movies arising from the film-editing guidelines. We will also show an application of discovering roles' social clusters using the automatically constructed social network.

The contributions of this work are summarized as follows: (1) we propose an automatic method to construct roles' social network from movies; (2) we present a new scheme to describe the interactions between characters; (3) we demonstrate how the automatically constructed social network can be used to discover social clusters in a movie. In the remainder of this paper, we start with a brief review of related works in Section 2. Then, Section 3 presents the automatic approach powered by a new scheme that quantifies the interactions among characters by the use of film-editing guidelines. We finally demonstrate the

[1] The term is defined in [10] where the words "character" and "role" are interchangeably used. Using "character" is more precise; however, we follow the same usage with [10].

effectiveness of the proposed approach, show its application, and conclude the paper with a short discussion summarizing our findings.

2. RELATED WORK

The roles' social network is, in general, described by a weighted undirected graph where vertices represent the characters in a movie and edges indicate the social closeness of two characters. Pioneering works applied face recognition and video processing techniques to identify each detected face and accumulate the count of the character appearance in each scene. For example, Weng *et al.* proposed to manually label the detected faces in the earliest *N* scenes to form the training data for face recognition [10]. The value of *N* is determined by the number of scenes in which all characters have appeared at least once. A scene detection method was applied to identify initial scenes. Manual labeling was again used to correct errors and get precise scene boundaries [10]. Wu *et al.* applied a similar approach that constructs social clusters from photo collections [12]. The co-appearance of individuals in photos is used to measure the social closeness of the involved individuals. These approaches make an assumption that people interact with others in a same physical place, and the strength of the connection only counts on the frequency that two people appear together, rather than the true interactions between them. Furthermore, these approaches are semi-automatic as the face recognition module requires human labeling of selected faces.

The analysis of social network has been proposed to facilitate a number of difficult visual tasks. For example, Wu *et al.* proposed to improve the performance of face clustering using the social relationship of people [11]. By assuming a person should have consistent connections to his/her neighbors, the social closeness likelihood is used as a feature to merge face clusters. In [10], the social network is used to determine the leading roles and roles' communication in a movie. The agglomerative clustering approach is applied to group characters into clusters based on their social activities. Liang *et al.* explored the characters' context in successive scenes to achieve precise scene segmentation in movies [5]. The social context has also been combined with face recognition scores in a conditional random field (CRF) model to label faces in consumer photos [8]. In this paper, we will show an application of discovering social clusters based on the automatically constructed social network.

3. APPROACH

In this section, we present a computational approach for constructing roles' social network from movies. The core of the approach is a new scheme that measures characters' social closeness based on their interactions, which can be effectively quantified using the film-editing guidelines. We start with the discussion of guidelines adopted in this paper, following a pre-processing procedure that forms face clusters, and, finally, present the method that constructs the roles' social network.

Fig. 1. Example of a storyboard designed by a director in picturing a scene.

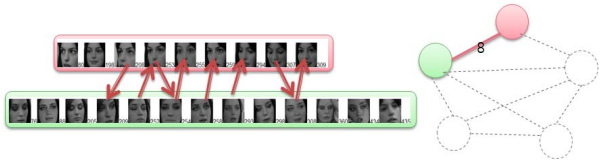

Fig. 2. Example of using shot alternation cues to quantify the social closeness of two roles.

3.1. Film-Editing Guidelines

In general, movies are edited based on a number of guidelines that aim at creating the perception of *continuity* across a shot cut [7]. These rules provide great information for grouping people by identity, as well as quantifying the magnitude of social activities between two individuals. We explore two rules in this work:

- The 180° rule suggests that a character will consistently appear on the left (or right) side of the screen through a scene when two characters are visible. In other words, relative position of actors stays constant.
- The shot alternation rule suggests that two consecutive shots usually show a different character before and after the cut, especially in a scene where a dialogue between two people is present.

Figure 1 shows an example of the storyboard in a scene of the movie "The Devil Wears Prada." Several cameras are settled to capture different views of the characters. By switching between the views with different angles, scene details can be highlighted so that the audience can better understand the story. Resulting from the 180° rule, *Emily* (the woman in shot 1) remains in the same horizontal portion of the picture. Also, consecutive shots take the frontal face of different people, e.g. *Emily* in shot 3 and

Andy in shot 4. The editing effects provide useful cues for grouping faces and constructing the social network in addition to low-level appearance-based features.

3.2. Forming face clusters

Firstly, we use the OpenCV library [14] to detect face regions in each frame and record the location and the scale information. Then, we compute the local binary pattern (LBP) [1] for face representation. Each face is described by a 59-dimensional feature vector—a histogram of 58 uniform patterns and one non-uniform pattern. The distance between two faces is measured by the chi-squared distance.

Based on the 180° rule, for a specific character in a shot, his/her face displacement and scale will not significantly change. We group detected faces into face tracks if faces in consecutive frames follow the constraints:

$$L_2(c_i, c_{i\text{-}1}) \le s_{i\text{-}1} \quad (1)$$

$$L_1(s_i, s_{i\text{-}1}) \le t \quad (2)$$

where c_i and $c_{i\text{-}1}$ are the coordinates of the current and the previous frames, and s_i and $s_{i\text{-}1}$ denote the scales. t is a threshold empirically set. In our example, we have $t = 0.2$. We remove face tracks if the number of faces in the track is less than 10 in order to filter false positive samples from the face detector.

Note that a character should have multiple face tracks after the face grouping step. We prefer separating the same character into different tracks, rather than mixing different characters in a track. Next, we select the face with minimum variances to other faces within a face track as the exemplar face to represent the whole face track. Therefore, the cost of computing the distance of two face tracks is reduced to the calculation of the chi-squared distance of two exemplar faces.

Finally, the affinity propagation clustering algorithm (AP) [4] is applied to group face tracks. The idea of AP is that data points—exemplar faces in our case—will continue to exchange messages until the system converges and produces a set of optimal cluster centers. A preference value is set to controls the number of data points selected as exemplars, with low preferences leading to few clusters, and vice versa. It is apparent that the average cluster purity, defined as the ratio of correctly assigned face labels and the total number of faces where each cluster is assigned to the most frequent label in the cluster, will decrease when faces are grouped into fewer clusters. As we aim to obtain a clustering result with the cluster number close to the number of characters in the movie, a small preference value is set, which will be discussed later in the experimental section.

3.3. Constructing Social Network

The social network is represented by an undirected weighted graph $G = (V, E, W)$ where $V = \{v_1, v_2, \ldots, v_N\}$ denotes the set of characters in a movie, $E = \{e_{ij} \mid \text{if } v_i \text{ and } v_j \text{ has relationship}\}$, and the element w_{ij} in W represents the strength of the relationship between v_i and v_j. The face clusters derived from the previous step constitute the vertices in V. Now, we introduce the computation of w_{ij} using the shot alternation cues.

We implement a simple shot change detection method to determine shot boundaries. Let S denotes the extracted shot sequence. Therefore, each detected face region f_i has two labels—cluster id $f_i.cluster$ and shot id $f_i.shot$. The interaction w_{ij} between two face clusters i and j is computed as follows:

$$w_{ij} = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}, \quad (3)$$

where m and n are the number of faces in the clusters, and a matrix $A = [a_{ij}]_{m\times n}$ is built with elements

$$a_{ij} = \begin{cases} 1, & \text{if } |f_i.shot - f_j.shot| \le 1, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

That is, the interaction between two characters is quantified by the amount of co-appearance and shot alternation between two face clusters. In the implementation, we linearly examine the cluster ids following the order of the shot sequence and adjust the values of W accordingly.

In Fig. 1, we show an example of a storyboard that contains two characters in a dialogue. Three cameras are settled to capture different views in the scene, resulting in multiple shots. The use of shot changes between two characters for quantifying their interactions is effective because shot alternation occurs frequently especially when characters are interacting with each other. Figure 2 illustrates two face clusters with the shot ids shown on the right-bottom corner of each face. An arrow is drawn if the face pair appears in the consecutive shots. We calculate the number of shot changes (8 in this example) and assign a weigh value of 8 for the corresponding edge in the social network.

Note that our method is not limited to dialog scenes where two characters are involved. In fact, our approach can generally deal with interactions that involve more than two people. More specifically, if a single shot has multiple characters A, B, and C, we create interactions A-B, B-C, and A-C. In consecutive shots, suppose shot t contains A and B, and shot (t+1) contains C and D, we create interactions A-C, A-D, B-C, B-D, A-B, and C-D. In the 1-to-n case where one character plays a dominant role, for example, shot t captures a speaker A and shot (t+1) captures his audience B, C, and D, the alternation approach creates interactions A-B, A-C, A-D, B-C, B-D, C-D.

The use of video editing cues has shown some success in person recognition [3]. However, the work in [3] focuses on the extraction of additional distances based on such cues to measure the dissimilarity between two faces. In this paper, we propose a completely different usage, aiming at the social graph construction.

In the implementation of shot detection, we impose a 7 by 3 grid on each frame. Each partitioned block is described by a histogram of pixel values quantized into 32 bins. The Bhattacharyya coefficient is used to measure the frame

similarity and a shot change is detected if the Bhattacharyya coefficient of two consecutive frames is below a pre-defined threshold.

4. EXPERIMENTAL RESULTS

We extracted a 36-minute video clip from the film—The Devil Wears Prada—, and detected 1,090 shots and 27,755 face regions using the OpenCV library [14]. These face regions belong to 14 characters and some of them are false positives. The AP clustering method returns 18 clusters with an average purity value 71.79%. The cluster purity values range from 52.17% to 100%. Four characters are merged with other characters during the clustering procedure. Also, we manually labeled each face in order to construct the ground truth data, which will be used to compare with the automatic approach. Moreover, the testing video contains 27 conversations, and the number of involved people ranges from 2 to 7. Figure 3 shows the distribution of the number of people participating in a conversation.

In the first experiment, we compare social networks built upon manually labeled faces using different approaches to describe the relationships among characters. The co-appearance based approach is used to construct the graph in Fig. 4 (a), while Fig. 4 (b) shows the proposed scheme based on shot alternation. The manually labeled social networks have 14 characters. The number on top of each face represents the character id. We firstly observed that two graphs have very similar structure while notable differences exist. For example, six edges—(2, 10), (2, 12), (2, 13), (3, 14), (4, 13), (10, 13)—are missing in Fig. 4 (b). The involved characters appear in the 12th scene of the movie, which captures a gathering of *Andy*'s college friends in a restaurant. Some interesting conversations happen between a subset of the people, but not all of them. It remains controversial as edges should or should not be built to connect every pair of the people in the group. Another interesting difference between the graphs is the edge (3, 11) is missing in Fig. 4 (a). The scene describes a phone dialogue between two people who physically appear at different places. The character no. 3 stays in a house and the character no. 11 walks on a street. We argue that an edge should be built to connect the two characters because they do have social communications. Similarly, the characters no. 5, 6, and 7 have stronger connections in Fig. 4 (b). These people appear in only a few scenes but the conversations last for quite a few minutes.

Next, we compare social networks established on manually labeled faces and the face clusters returned from the affinity propagation clustering approach [4]. Figure 4 (d) shows the social network derived by the automatic approach that has 18 clusters with 10 distinguishing characters. The number below each face represents the cluster id. Four characters (no. 8, 9, 11, 14) are missing in the construction process while three characters have multiple clusters. For example, faces of the main actress (no. 1) are grouped into 6 clusters (1, 2, 3, 5, 12, 13).

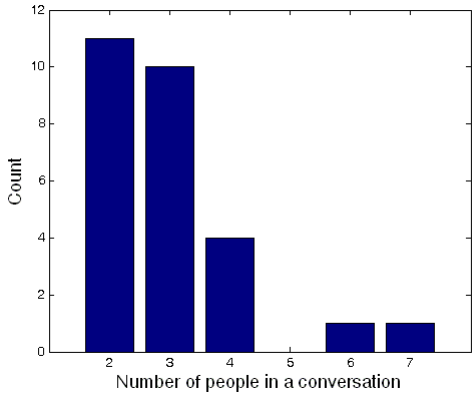

Fig. 3. Distribution of the number of people participating in a conversation.

The missing characters in the automatically generated social network (Fig. 4 (d)) are inevitable because these characters appear just a short time in the movie, and, thus, their face tracks account for only a small portion compared to those of other characters. Therefore, their face tracks tend to be combined with other characters in the clustering process. For example, the characters no. 8, 11, 4, and 13 all wear glasses and have less hair covered their forehead. The characters no. 8 and no. 11 are merged with the characters no. 4 and no. 13 due to the similar appearances. Similarly, faces of the leading roles are divided into more than one cluster because of the wide range of variations in pose and facial expression.

Although the automatically generated and the manually labeled networks look differently at the first glance, they somehow have a similar structure. For example, if we simply merge the clusters that contain a same person, the refined social network (Fig. 4 (e)) is approaching the manually labeled one (Fig. 4 (b)). If we further remove the four missing roles from both social networks, shown in Fig. 4 (c) and Fig. 4 (f), respectively, we can observe that these graphs are quite similar. Only four error edges—(4, 7), (4, 12), (2, 10), (10, 12)—remain in the automatic constructed graph, caused from the clustering errors where a few different characters are mixed into a same cluster. The proposed approach generates similar social graphs from either the manually labeled faces or the automatically clustered face tracks.

Finally, although only one movie was used in the experiment, we argue that the interactions of characters should not be significantly divergent given different genres of films. The "misc en scene" process—characters enter a space and interact with each other to narrate a story segment—is general.

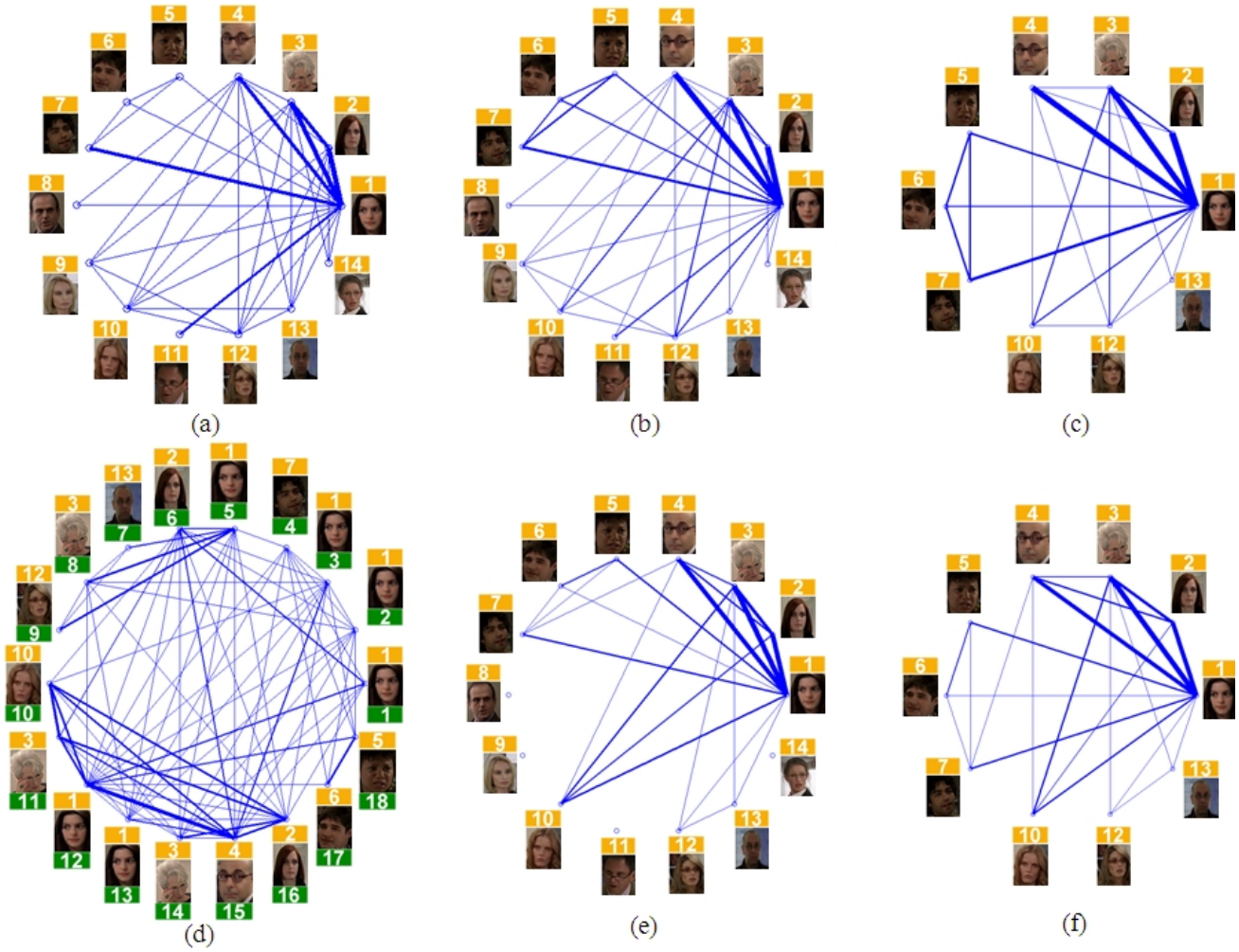

Fig. 4. Roles' social networks of the movie "The Devil Wears Prada": (a) manual approach based on co-appearance [10]; (b) manual approach based on interaction; (c) simplified graph by removing roles no. 8, 9, 11, and 14 from (b); (d) the automatic approach based on interaction; (e) refined graph from (d) by merging nodes of the same role; (f) reduced graph of (e) by removing four isolated nodes. Please see text for details.

5. APPLICATION

Finally, we demonstrate an application of discovering characters' social clusters powered by the social network. We consider a social cluster as a set of people, and any two people in the cluster should have interactions with each other. Unlike previous methods that apply bottom-up agglomerative clustering techniques to group identities into clusters [10][12][13], we formulate a maximal clique problem given a social graph. Thus, the social clusters can be naturally discovered by finding the maximal cliques in the graph.

Since the social graph is undirected, we apply the Bron-Kerbosch algorithm [2], which is a recursive backtracking algorithm that searches for all maximal cliques given a graph G. The worse-case running is $O(3^{n/3})$ [2], where n is the number of vertices. Since the number of characters in a social graph is quite limited, the computation for searching for all maximal cliques is efficient. Figure 5 shows the discovered social clusters where seven maximal cliques are identified using the Bron-Kerbosch algorithm. Characters are neatly separated into several groups based on their social activities. For example, the leading actress's college friends (no. 5, 6 and 7) form a social group, separating from her colleagues. It is interesting to observe that the figure delivers information consistent with those perceived by people after seeing the movie to certain extent.

6. CONCLUSIONS

We propose an automatic approach to construct roles' social network from movies. In particular, a new scheme that

measures the interactions among characters in the network is introduced. The method is conceptually simple, easy to implement, and requires only the shot boundary information to fairly describe characters' interactions. We also show the discovery of social clusters from a social network. However, we should point out the approach is applicable for most modern movies where the continuity editing assumption holds. For "French New Wave" types of movies where pieces of films have nothing to do with the overall story, or those movies which contain no (or just a few) interactions among characters, the gain of the proposed scheme would be limited.

There are a few directions we may explore. For example, one research direction we are pursuing is the use of social network constructed by our method for face annotation. Our method produces fairly meaningful information that describes characters' social interactions from a noisy face clustering result. This network should provide complementary cues to appearance-based features for recognizing faces of an identity. Another research direction is the exploration of a metric for evaluating the social networks. The quality of social networks is usually measured through the community analysis or the performance of story segmentation [10]. It is not clear which graph similarity notation truly reflects the social network similarities. Finally, we will conduct extensive experiments using more movies and provide deeper analysis on the automatically built social graphs.

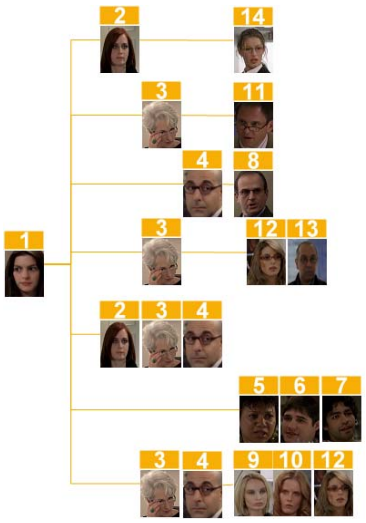

Fig. 5. Discovered social clusters in a social network.

6. ACKNOWLEDGEMENTS

This work was supported in part by the National Science Council, Taiwan, under Grants NSC 100-2631-S-003-006 and NSC 99-2221-E-003-027.

7. REFERENCES

[1] T. Ahonen, A. Hadid and M. Pietikäinen, "Face description with local binary pattern: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037-2041, 2006.

[2] B. Coen and K. Joep, "Algorithm 457: finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, pp.575-577, 1973.

[3] T. Cour, B. Sapp, A. Nagle, and B. Taskar, "Taking pictures: temporal grouping and dialog-supervised person recognition," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2010.

[4] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, pp. 972–976, 2007.

[5] C. Liang, Y. Zhang, J. Cheng, C. Xu and H. Lu, "A novel role-based movie scene segmentation method," in *Proceedings of Pacific-Rim Conference on Multimedia*, pp. 917-922, 2009

[6] M. Plantie and M. Crampes, "From photo networks to social networks, creation and use of a social network derived with photos," in *Proceedings of ACM International Conference on Multimedia*, 2010.

[7] T. J. Smith, "An attentional theory of continuity editing," PhD thesis, University of Edinburgh, 2005.

[8] Z. Stone, T. Zickler, T. Darrell, "Autotagging facebook: social network context improves photo annotation," in *IEEE Workshop on Internet Vision*, 2008.

[9] A. Vinciarelli and S. Favre, "Broadcast news story segmentation using social network analysis and hidden Markov models," in *Proceedings of ACM International Conference on Multimedia*, 2007.

[10] C. -Y. Weng, W. -T. Chu, and J. -L. Wu, "RoleNet: movie analysis from the perspective of social network," *IEEE Transactions on Multimedia,* vol. 11, no. 2, pp. 256-271, 2009.

[11] P. Wu and F. Tang, "Improving face clustering using social context," in *Proceedings of ACM International Conference on Multimedia*, 2010.

[12] P. Wu and D. Tretter, "Close & closer: social cluster and closeness from photo collections," in *Proceedings of ACM International Conference on Multimedia*, 2009.

[13] K. Yuan, H. Yao, R. Ji, and X. Sun, "Mining actor correlations with hierarchical concurrence parsing," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010.

[14] *Open Source Computer Vision Library*. Available: http://www.intel.com/technology/computing/opencv