Social Network Inference in Videos

Ashish Gupta, Alper Yilmaz

The Ohio State University
Columbus, OH 43210, United States

Abstract

Video analysis with the aim of discovering social relations between the people in that video is an important and unexplored topic with significant benefit towards a higher level understanding of videos. This article focuses on the inference of two social groups in each video where members of each group share friendly relations with each other and have an adversarial relation with members of the other social group. Using low-level audio-visual features and motion trajectories we compute a measure of expression of social relation in each scene in video. The occurrence of actors in each scene is computed using face recognition with Local Binary Pattern descriptor. The actor-scene forms a 2-model social network, which we use to compute a 1-mode network of actors. The leaders of each group, which are the actors with greater social impact are estimated using Eigen-centrality. We demonstrate our approach on several Hollywood films, which span genres of action, adventure, drama, sci-fi, thriller, historic, and fantasy. This approach is successful at using video content analysis to infer the two social groups and typically the principal protagonist and antagonist in the films as well.

Keywords: Social Network, Video Analysis, Face Recognition, Gaussian Process, Video Segmentation, Optical Flow, Graph Centrality

Email addresses: gupta.637@osu.edu (Ashish Gupta), yilmaz.15@osu.edu (Alper Yilmaz)

Preprint submitted to Elsevier *March 29, 2017*

Social Network Inference in Videos

Ashish Gupta, Alper Yilmaz

The Ohio State University
Columbus, OH 43210, United States

1. Introduction

Video analytics has experienced a lot of progress in the past decade. Nevertheless, the progress made in video understanding, that of extracting high-level semantics in the form of relations between people in a video, is in comparison, insufficient. The explosive growth in social media makes this an interesting task. In this article, we discuss an effective approach towards learning interactions between people, building social networks, inferring social groups, and discovering the leader of each of these groups in a video from a sociological perspective. Our principal contributions include inferring the associations between low-level visual and auditory features to social relations by utilizing machine learning algorithms such as, support vector regression and Gaussian Processes. The inferred social network is subsequently analyzed to: discover communities of people; and identify a leader of each community. These are arguably two of the most relevant objectives in social network analysis. Additionally, as an extension to the basic framework, we discuss the relationship between visual concepts and social relations that have been explored in [1]. Herein, the visual concepts can be considered as mid-level visual representation in inferring social relations and these are then compared with features used in the basic framework.

The analysis of scene from video has been a subject of intense research. Researchers want to understand the scene by analyzing patterns in motion. The motion is estimated by computing the trajectories of various objects in the scene [2, 3, 4, 5, 6]. However, the majority of these research efforts limited in their ability to infer semantic information from video. They conduct elementary analysis, like clustering of motion trajectories or understanding actions of tracked objects

Email addresses: gupta.637@osu.edu (Ashish Gupta), yilmaz.15@osu.edu (Alper Yilmaz)

[7, 8, 9, 10], where each object is tracked separately from others. They do not focus on trying to understand some manner of group behavior in the pattern of multiple trajectories. Broadly speaking, computer vision researchers have not investigated the content in video from the perspective of sociological relations between entities in that video. This would have provided an understanding of the actions of people in terms of their mutual relations. Within the purview of existing research work on action recognition and analysis, the use of inferred social relations used in conjunction with other visual features can potentially aid in the disambiguation of complex events in video, by providing relevant contextual information.

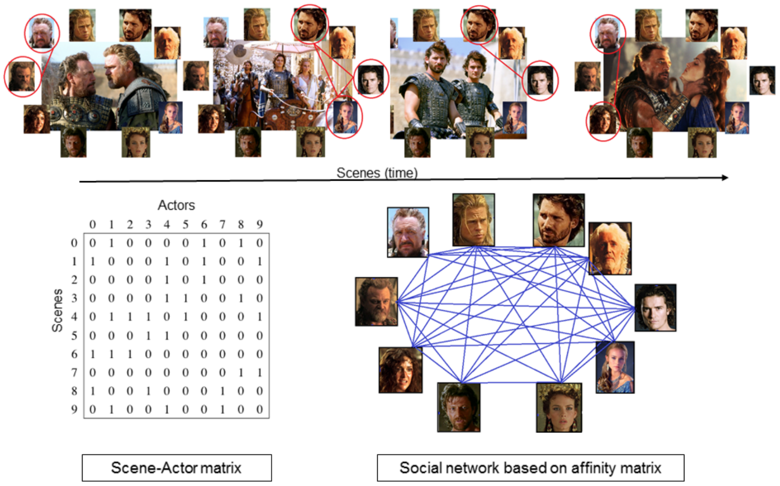
Social networks have evolved to become the data structure of choice to represent and aid in the analysis of social relations between various entities in a sociological context. Social relations are modeled by a network or a graph structure, consisting of nodes and edges. The nodes represent individual entities within the network, and the edges denote the relations between these entities. When analyzing video to infer social relation between people in that video, the entities are the actors or characters. The graph built from analyzing a video now serves as a basis to discover social groups in that network. A social group is also referred to as a community, wherein members of a community share a common mutual social relationship. These communities are typically discovered using the connectivity between the nodes in the graph, which are the actors in the video, using social network analysis algorithms. The modularity algorithm [11] is a popular example. Social network analytics has also drawn attention from research communities in data mining[12, 13] and content analysis of surveillance videos [14].

In order to understand and be able to model complex social relations between people in video, we turned to feature films. Such films provide plenty of scenarios for social interaction between various characters in that video. In addition to this, visual and auditory features are always available. Since a film is continuous video across different contextual situations we must first segment the film to minimize contextual mixing. The film is segmented into shots and scenes [15]. Then actors occurrences in scenes is detected using face recognition [16]. The characteristic of our framework that makes it unique is the ability to apply it to a social network, which have adversarial relations in addition to regular associations between actors. This is a significant topic in sociology that has not received adequate attention. This perhaps is in part due to the inherent challenge of simultaneously modeling adversarial relations in conjunction with friendly relations. We consider an adversarial social network to constitute two distinct communities. The actors in each of these communities have a friendly social relation with other actors in

Figure 1: Social communities in the film Troy. These are rival communities with their respective group of actors and a leader. We automatically detect these two rival communities (Greeks and Troyan), and also identify the respective leaders (Achellies and Paris)

the same community and also have adversarial relation with actors in the other community.

We employ visual and auditory features in our framework towards quantitative analysis of potential clustering of actors in each scene. This provides a soft constraint between the actors. These soft constraints are subsequently combined to compute an affinity between pairs of actors. In this social network, communities are computed using a generalized modularity principle on the inter-actor affinity matrix [17]. Next we explore the concept of leadership. Social communities tend to have a central figure, a leader or a hub. This is a person that has the strongest connectivity to most of the other people in that community. Consequently, this person is also the most influential. Arguably, we consider this influential person as the leader of that community. We identify the leader using Eigenvector centrality. To illustrate the point, Figure 1 shows two communities for the feature film, Troy, the Greeks and the Troyans. Each actor in each communities contributes to the group, but certain actors, Achellies for the Greeks and Paris for the Troyans are the most influential, and their social interactions are the the most important in that film, making them community leaders, from a social network perspective. Our learning based framework for computing social networks is shown in Figure 2. In this example, we consider the film Troy. There are 10 actors that we will use to build the social network. As we process the frames of the movie sequentially, we analyze each frame for the occurrence of one or more of the 10 actors. Actors that co-occur share a social relation by virtue of presence in the same scene. This social relation can be friendly or adversarial. We record this social relation by scoring the appropriate scene-actor matrix. After the entire movie is processed,



Scenes \ Actors	0	1	2	3	4	5	6	7	8	9
0	0	1	0	0	0	0	1	0	1	0
1	1	0	0	0	1	0	1	0	0	1
2	0	0	0	0	1	0	1	0	0	0
3	0	0	0	0	1	1	0	0	1	0
4	0	1	1	1	0	1	0	0	0	1
5	0	0	0	1	1	0	0	0	0	0
6	1	1	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	1
8	1	0	0	1	0	0	0	1	0	0
9	0	1	0	0	1	0	0	1	0	0

Figure 2: Social network graph of actors in film Troy (2004) is constructed from the affinity matrix of film-scenes and actors.

we use to the scene-actor matrix to compute a affinity matrix between actors. Those actors that occur in the same scene naturally share a stronger social affinity as compared to two actors who rarely ever co-occur. The graph between actors where the edge attribute of the graph is based on the affinity matrix is later analyzed to infer social relationships. The rest of this article is organized as follows. We begin with a review of related work in Section 2. Our approach to video shot segmentation is explained in Section 3. The computation of the scene-actor occurrence matrix using face recognition is described in Section 4. Next we describe our approach to use low-level visual and auditory features to compute information for grouping in Section 5. In section 6, we detail how social networks are learned from videos. In Section 7, we describe our methodology used to analyze these social networks. We evaluate our approach on a set of films in Section 8. The association between visual concepts and social relations are explored in Section 9. We summarize this article in Section 10.

2. Related Work

The ideas in social networking is not new or isolated to the field of sociology. In one form or another it has been used in other fields to solve various problems in their respective domains. These fields include data mining, computer vision and multimedia analysis [12, 14, 18]. In this section, we elaborate on various related approaches and where they sit in relation to our problem domain.

Understanding relations between people in surveillance video is important towards predicting behavior. Towards detecting such possible groups of people, the work in [14] employs traditional methods in social network analysis. This paper conjectures that the interactions between people is based on their physical proximity. This conjecture, that physical proximity is correlated to social proximity is not guaranteed. Nevertheless, it provides a quantitative measure of social relations between actors in a video. It uses traditional modularity [11] to detect potential groups. In [19], the authors extend this measure of social proximity based on distance to include relative velocity between various tracked objects. They next use clustering to discover groups in a crowd, wherein people belonging to the same group, while inter-mixed with other people will tend to have correlated velocity vectors.

The work in [18] generates a social network from co-occurrences of people in the video. The authors do this without attributing them to some low level video features. In their work, the relations are limited to be only friendly relations. This means there can only be one social group with various degrees of affinity between various actors. Such a single group can be easily detected using regular clustering on the graph structure. This co-occurrence concept correlates to our framework, however, we extend further to relate them to visual and auditory features. Importantly, we utilizes these features to quantify the types of social interactions between various actors and communities.

Some of the recent work in computer vision, including for example that in [20], has tried to identify interactions between objects in terms of categories from video using Markov Random Fields (MRF). However, their objective isn't estimating social relationships for a group of individuals. In a similar vein, the work in [21] delves in the task of identifying social roles played by actors in a scene using weakly-supervised Conditional Random Fields (CRF). However, these authors did not leverage social network structure for their analysis. Some research pursuits in the field of data mining have recently explored approaches to mining relations in a network using log-entries as raw data. In [12, 22] the research groups have analyzed social networks and their dynamics using Bayesian model-

Data source	**Observed features**	**Construction method**	**Usability**	**Examples**
From interaction logs	Social interactions	Simple connections	On collected social data, e.g. emails	[12, 22]
From cell phone usage	Call data, etc.	Simple connections	On collections of mobile devices	[23, 24]
From videos (existing)	Tracked people	Proximity heuristics	On surveillance videos	[14, 19]
From videos (this chapter)	Audio-visual cues	Learning approaches	On videos with training labels	[1]

Table 1: Our framework in context of related work on inferring social network relations.

ing of social networks and the interactions among actors. Similar to the examples in the data mining field, other researchers have also utilized data on cellular phone usage to estimate social networks[23, 24]. We note that these set of methods do not make use of visual data. Videos are a very rich source of information and are being increasingly adopted for communication in society. The use of videos to infer social relations is both viable and valuable towards social network analysis. By employing audiovisual features in video our basic framework creates a new avenue for understanding social interactions in videos. Comparatively speaking, it is relatively simple to utilize network log-entries and mobile phone usage, to extract social information. However, video data represents a significant challenge toward inferring social information using pattern recognition techniques.

The principal novelty in our framework and its difference from the research work we reviewed here and other works in the domain of sociology is the approach that we have adopted to address the problem of inferring a social network. In particular, the existing techniques define social interaction heuristically and construct a social network using these heuristics. In contrast, our approach is analogous to human perceptual approach to inferring social relations in an observed scene. Imagine a scene wherein there are numerous actors involved in some activities. A human observer, bereft of prior knowledge of the context of that scene would assume all actors to be equivalent and similarly related to others. In other words, there is no bias in any pair of actors. As the scene plays out, the human will begin conjecturing the affinity between pairs of actors and using this to roughly construct a social network. Analogous to human intuition , we observe interactions and learn communities in the network from observations in a scene without prior bias. It is our conviction that this approach will benefit other application domains in computer vision. For example, automated analysis of video of a professional meeting, where modeling the actions of individuals an inferring high level semantic relations between the attendees is important [25, 26]. We have summarized some of these works in Table 1.

3. Video Shot Segmentation

In order to record occurrences of actors and compute audio-visual features, we must first divide the film into shorter segments. Ideally, a video shot corresponds to the view from one camera of the same set of actors. When there is a change of camera, even though there is no change of scene, we consider it a change in shot. For example, consider a dialogue between two actors. In the first shot, the camera is behind actor A and shows his back along with showing the face of actor B. Note that the camera may move a little, zoom in or out, and maybe pan a little, but so long as the view does not introduce a new actor and actor B is continuously visible, then there is no change in video shot. If the video next shows the view from camera behind actor B and the face of actor A, then there has been a shot transition at this point, since the actor whose face is visible has now changed. This will be reflected in the scene-actor 2-mode social network. A film scene typically consists of several shots, wherein the camera view alternates between two actors in a dialogue. It also includes the scenario where the camera slowly pans or zooms out and thereby introduces new actors to the view. The principal factor that defines a shot transition is significant change in scene content. We measure this using dense optical flow. However, since motion of actors within a shot or small motion of camera itself will also amount to a perceptible change in optical flow. Therefore we empirically determine a threshold on cumulative motion across frames of the video. Figure 3 shows the cumulative motion value for frames in a video of film Troy. Note that a low threshold allows for large motion within a single shot to be segmented into multiple shots. This is undesirable since large motion sometimes occurs in films with a busy scene where multiple actors are simultaneously moving, like a dance floor scene or a battle scene. The Hue-Saturation-Value (HSV) color based score for each frame, shown in the top of Figure 3 can also be used to disambiguate between false-positive shot transitions, since the HSV values would typically change when the shot changes.

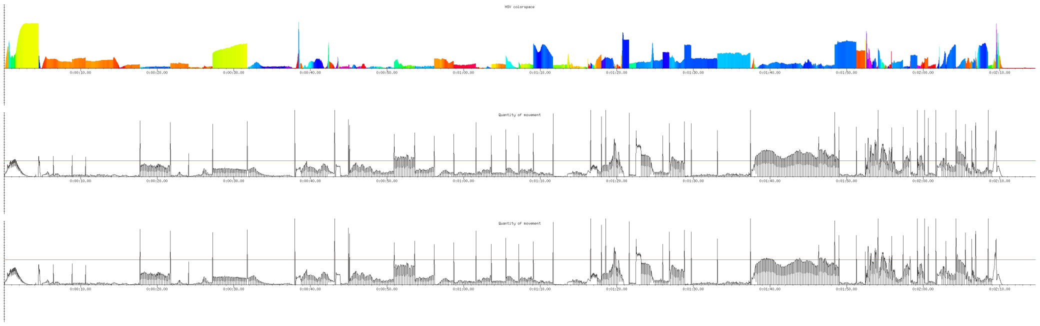


Figure 3: Video segmentation into shots based on a cumulative measure of activity in the video. At the point of transition between shots, there is a huge variation in visual content of frames, since different video shots have different foreground and background. Since there is activity within a single shot and sometimes the different in background between shots may be very small (background is sky), we empirically determined a activity threshold (shown by the red line). Therefore, the time at which video activity greater than this threshold is marked as a shot transition point.

The results of our shot segmentation approach is shown in Figure 4. We show

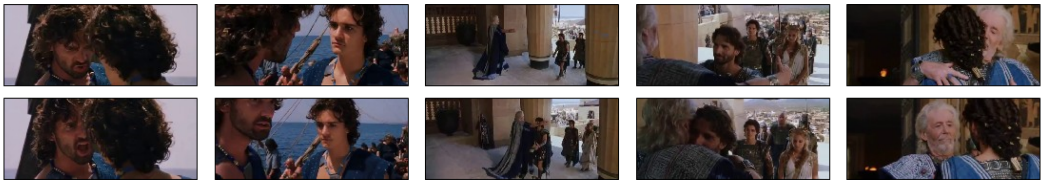


Figure 4: Video segmentation into shots. Figure shows sequence of shots from a scene in film Troy. The images in the top row are the first frame in each segmented shot and the images in the bottom row are the last frame in that shot.

the first frame of each shot in the top row and the last frame of that shot in the bottom row. We are successful as segmenting the video into shots where each shot has unaltered set of actors engaged in some activity and the transition frame is properly identified.

4. Actor Recognition

In order to compute a scene-actor 2-mode social network, we detected and recognized faces in the video frames. Instead of recognizing the entire body of an actor we chose to restrict recognition to the actor's face only. This choice was based on the intuition that a typical actor appears in multiple different costumes and thereby the only consistent visual feature is the actor's face. We used the Local Binary Pattern (LBP) cascade descriptor to detect candidate face regions in a frame. We also used LBP for modeling a face. Typically, face recognition works well when we have a full frontal image of the face. Changes in pose towards a profile pose makes accurate face recognition challenging. The use of costumes that partially occlude the face also add to the challenges. We acquired the training image for each actor's face from Google Image Search engine. We queried the search engine with three types of queries for each actor. One is the name of the movie and the name of the actor, second is the name of the movie and the name of the character played by the actor, and third is just the name of the actor. These sets of downloaded images were collated into one raw image set. We used the LBP detector to identify candidate face bounding-boxes in the raw image set. A single image can have one or more instances of a face in them. The cropped bounding-boxes in all images were resized for consistency. Since the LBP cascade detector returns several false-positive faces images, we introduce a further pruning step to remove images that did not have a face in them. We converted the resized images to a YCrCb color-space which work well for skin color based operation. We detected contours in the bounding-box. A contour corresponding to a human face

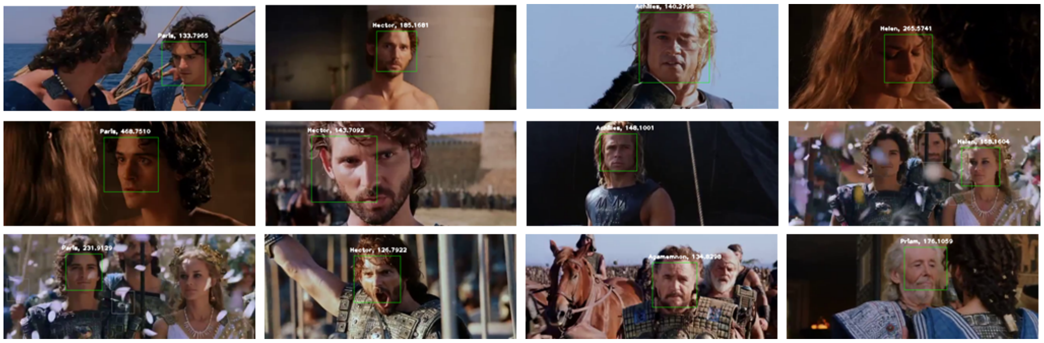


Figure 5: Detecting actor occurrence in video frames. Figure shows examples of face recognition of the actors in film Troy. Each image shows one or more bounding-boxes centered on a detected face in that video frame. The name of the most likely actor and associated confidence score is shown above the box.

can be expected to have about three-fourths the area of the bounding-box itself. We used this heuristic to prune those images which were false-positives.

We trained our face recognition classifier, where each actor was considered as a class. The face recognizer associates a confidence score with each actor. The actor with the high score is the predicted label. We show examples of our face recognition implementation for frames in film Troy in Figure 5. In the figure, it is clear that our face recognition can handle small changes in pose, partial occlusion, large changes in background and various ambient lighting conditions.

Since large changes in pose lead to misclassification, we pursue an aggressive pruning strategy where only those bounding-boxes with a confidence score higher than an empirically determined threshold value are retained. Since we building a scene-actor occurrence matrix, we only require one reliable detection anywhere in a scene. As a scene is composed of multiple shots and a shot is composed of multiple video frames, we can afford to commit Type I statistical error but not Type II error[1].

5. Learning to Group Actors

We begin with the condition where there is no known relation between any actor in the video. We only have low-level visual features available to us. Since

[1] In statistical hypothesis testing, a type I error is the incorrect rejection of a true null hypothesis (a ”false positive”), while a type II error is incorrectly retaining a false null hypothesis (a ”false negative”).

there are no discernible social communities, we can not explicitly label any community. We extract low-level visual features from videos. The kernels on features at scene level provide us criteria for grouping actors using regression we have learned from other videos. Unlike several other approaches, our mapping strategy is data-driven and provides a flexible and extensible approach. This approach can incrementally use new features as they are observed in the video.

Say the video $\mathbb{V}$ we consider is composed of scenes, $s_1, s_2 \cdots s_M$, each of which contains a set of actors and has an associated grouping criteria $\Gamma_i \in [-1, +1]$. The grouping criteria is used to decide the affinity between actors who occur in the same scene. These two actors belong to the same community if ($\Gamma_i > 0$) or different communities if ($\Gamma_i < 0$). The absolute value of Γ_i is indicative of the definitive association of actors and community. Next, we detail our approach on estimating such grouping criteria from low level features observed in a video.

We proceed with the assumption that there is a difference in the nature of interactions that occur between a pair of actors belonging to the same community as compared to the interaction between pair of actors from different communities. This conjecture facilitates an anonymous grouping of actors into their respective communities, since the difference in interaction is a function of their community membership rather than any property unique to an actor. This serves as a weak grouping criteria. As a consequence, any inferred community labels from one set of training videos can be propagated to a novel video. It follow that actions of actors in a video described by low-level audiovisual features are positively correlated to the type of social relationship among the actors. In this context, we define an activity as an expression of social relationship. Stated in another way, the association between activities and actors provides a distinct feature set that can be used to infer if members of a single community or different communities co-occur in the same scene. For example, tutors and students in a video in a school can be observed to interact in distinct manner within and across the two communities. Put simply, low-level audiovisual features in a scene are correlated with the predominant social expression in that scene.

While the gamut of social interactions is very large. The underlying conjecture in this work is that the nature of activities between two antipodal communities is sufficiently distinct to allow for effective separation of actors into two distinct groups. The friend and foe interactions are encoded using low-level features into friendly and adversarial relations.

We define a scene as the smallest segment of a video such that one scene contains one event, were an event is expression of one social relation. In other words, an event can be considered as associated with expression of either friendly or ad-

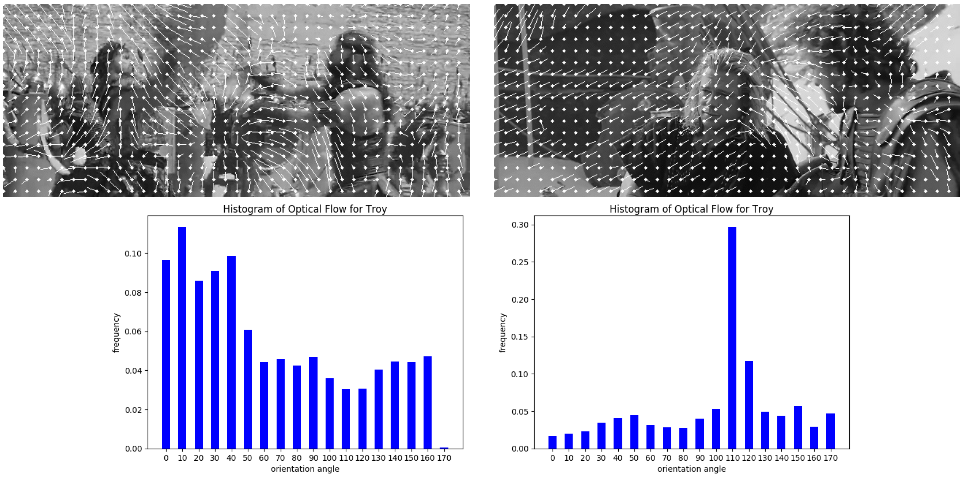


Figure 6: Social interaction is correlated to optical flow pattern in video. Figure shows examples of scenes from film *Troy* with dense optical flow in a sample frame and its associated histogram of orientations of motion vectors.

versarial relationship. Consequently, low-level features generated from the video and audio of each scene can be used to quantify adversarial and friendly scene features. To help us disambiguate, we note that film directors typically follow certain structure for conveying a story and dramatic content. This is referred to as cinematic principles in film literature, and is to emphasize the adversarial content in scenes for dramatic effect. Accordingly, adversarial scenes have sudden and surprising changes in visual and auditory contents, which are effective in conveying an atmosphere of conflict and tension. In sharp contrast to this friendly scenes have gradual change in visual and auditory content to convey a sense of calm and cooperation. Therefore, the visual and auditory features, which quantify friendly/adversarial scene content, can be extracted by analyzing the audiovisual disturbances in a video [27].

5.1. Visual Features

To measure visual disturbance, we formulate our interpretation of cinematic dramatic principles in terms of motion field from all actors in a scene. Regardless of actor identity and specific social interaction between various actors, in an adversarial scene, the motion field is distributed in multiple directions. We illustrate this with an example in Figure 6. We compute the motion field using optical flow distribution. We use dense flow field generated by estimating optical flow at each pixel [28] in the frames extracted from the video. The other option is to use

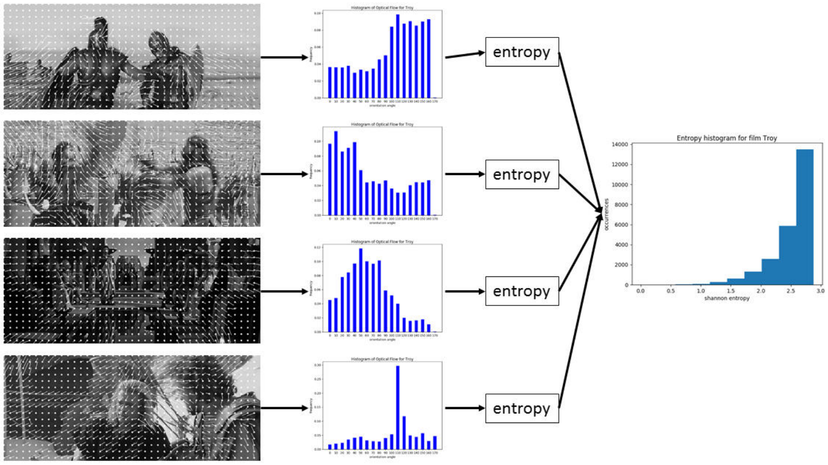

Histogram of Optical Flow for Troy
frequency
orientation angle
entropy
Histogram of Optical Flow for Troy
entropy
Histogram of Optical Flow for Troy
entropy
Histogram of Optical Flow for Troy
entropy
Entropy histogram for film Troy
occurrences
shannon entropy
0
2000
4000
6000
8000
10000
12000
14000
0.0
0.5
1.0
1.5
2.0
2.5
3.0

Figure 7: Computation of histogram of entropies of frames from video. Figure shows example of film Troy. Based on social interaction, each frame has its own entropy. The histogram of these entropies for all frames in the film Troy is shown on the right. For comparisons we utilize normalized histograms.

specific points which are have relevant visual information in their local neighborhood. It should be noted that typical tracking implementations use ‘good features to track’ approach. This approach works well for tracking few objects in a relatively uniform background. However, the degree of background activity in films is high since the camera view point keeps changing. Therefore we use a dense flow field with lower memory to reduce the erroneous effects of camera movements and change in view points. The example on the left in Figure 6 shows a adversarial scene, where Hector and Ajax in Troy are fighting. Note the orientation histogram and the quiver plot of motion flow fields. Several bins of the orientation histogram have high values. In contrast to this, note the example on the right of Achilles and Odysseus talking, which have motion flow field in harmony and the orientation histogram is unimodal.

Next, we need to quantify the flow field, specifically, the disturbance in the flow field. The entropy of the orientations of the flow field vectors provides a good aggregate measure of the degree of disturbance in the scene. As we stated previously, an adversarial scene will have a more isometric distribution of flow field, which translates to a higher entropy of the histogram of orientations. In a friendly scene, the actors will typically be moving together, which means the flow field will be unidirectional and the histogram of orientations will be sparse with a unimodal distribution. The entropy of this histogram will be lower than that

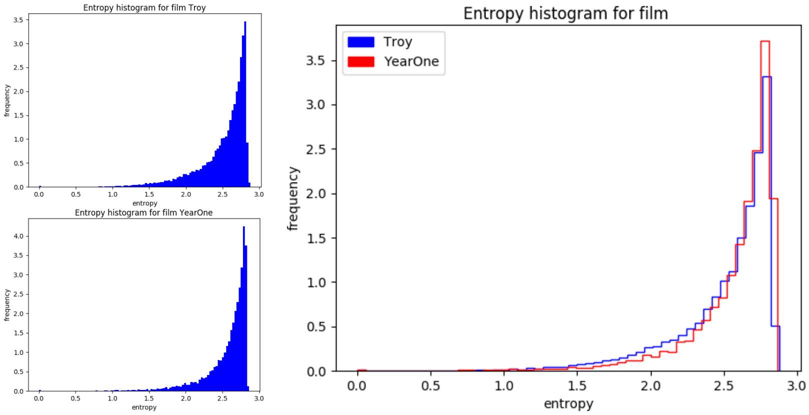


Figure 8: Entropy histogram of films *Troy* and *Year One*. The aggregate entropy histograms of these two different kinds of films is fairly similar. Consequently, a normalized entropy histogram is a descriptor that is unbiased by the film itself. It can therefore be used consistently for different videos without the need to modify the descriptor based on the film.

for isometric orientation distribution of adversarial scenes. This simple but highly effective idea is illustrated using examples in Figure 7.

The computed histograms of optical flow vectors are weighted by the magnitude of motion. The lower valued bins of the entropy histogram correspond to scene with unidirectional motion and the higher valued bin corresponds to scene with isometric directional motion. Our results corroborate our conjecture as is evident in Figure 9. The flow distributions in adversarial scenes tend to be isometrically distributed and thereby consistently have a bias towards higher entropy bins as compared to friendly scenes. We now have an established criteria for distinguishing between friendly/adversarial scenes.

5.2. Auditory Features

We extract the audio track in the film. Therefore, for each scene we have both the video and its corresponding audio track. Typically, there is correlation between nature of audio and visual disturbance in a scene. So adversarial scenes will typically contain more dramatic sounds. In keeping with the atmosphere of tension in such scenes, the voices of actors will be at a higher pitch. Consequently, auditory features extracted from the audio track can be used in conjunction with the visual features to improve the performance of our framework. To encode relevant properties of the audio track we use features in both time and frequency domain. We utilize the types of auditory features discussed in [29, 27]. The features we consider in this article are:

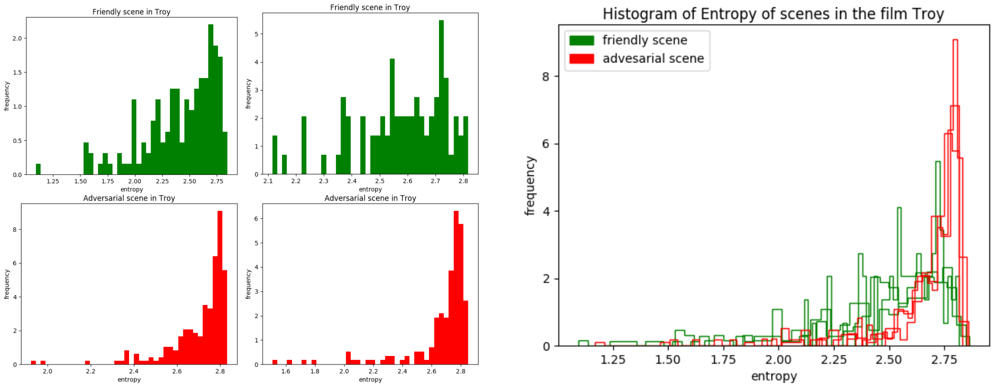


Figure 9: Entropy histogram based description of friendly and adversarial social relationship in video. The top-left histograms in 'green' show the entropy histogram of scenes in 'Troy' with friendly social interaction. The bottom-left histograms in 'red' pertain to scenes with adversarial social interaction. It is fairly evident that the histogram shape is correlated to the social interaction. For comparison, the histogram on the right shows overlay of histograms from 3 friendly and 3 adversarial scenes in Troy. The friendly scene histograms have a bias towards lower-entropy bins, whereas the adversarial scenes have a bias towards high-entropy bins.

1. Energy Peak Ratio $EPR = \frac{p}{S}$, where p is the number of energy peaks and S is length of an audio frame
2. Energy Entropy $EE = -\sum_{i=1}^{K} e_i \log e_i$, where a audio frame is divided into K sub-windows. For sub-window i, energy e_i is computed
3. Short-time Energy $SE = \sum_{i=1}^{S} x_i^2$, where S is the length of an audio frame
4. Spectral Flux $SF = \frac{1}{KF} \sum_{i=2}^{K} \sum_{j=1}^{F} (\varepsilon_{i,j} - \varepsilon_{i-1,j})^2$, where $\varepsilon_{i,j}$ is the spectral energy at sub-window i and frequency channel j
5. Zero Crossing Rate $ZCR = \frac{1}{2S} \sum_{i=1}^{S} |sgn(x_i) - sgn(x_{i-1})|$, where sgn stands for a sign function

We compute these features for sliding window that is 400 ms in length. The value of each feature that is used to populate the feature vector is the average of the corresponding features for the duration of the scene. An example of these auditory features is shown in Figure 6 for both friendly and adversarial scenes. The adversarial scenes have more peaks as compared to friendly scenes.

5.3. Grouping Criteria

We compute visual and auditory features for every scene. We quantize the features to 5 bins, and so we have two vectors per scene: a 5-dimensional visual feature vector and a 5-dimensional auditory feature vector. We use both of these for each scene to compute a grouping criteria $\Gamma_i \in [-1,+1]$ for that scene s_i. Towards this we use Support Vector Regression (SVR). Specifically, we use radial basis function for both the visual and auditory feature vectors, which leads to two kernel matrices $\mathscr{K}_v$ and $\mathscr{K}_a$ respectively. The bandwidths of these two kernels is determined using cross-validation. The joint kernel is the product of these two kernels, given as

$$\hat{\mathscr{K}}(u,v) = \mathscr{K}_v(u,v)\mathscr{K}_a(u,v)$$

The dual criteria in our SVR implementation is to find a function $g(\cdot)$ that has a maximum deviation of ε from the labeled targets for the training data and also is as flat as possible. The decision function can be written as,

$$\Gamma_i = g(s_i) = \sum_{j=1}^{L} (\alpha_j - \alpha_j^*)\hat{\mathscr{K}}_{l_j,i} + b \tag{1}$$

where the coefficient b is offset, α_i and α_i^* are the Lagrange multipliers for labeling constraints, L is the number of labeled examples, and l_j is the index for the j^{th} labeled example.

In our problem domain,the joint kernel together with training video scenes and their grouping criteria $\Gamma_i = +1$ (scene with members of only one community) and $\Gamma_i = -1$ (scene with members from different communities) leads to grouping constraints for a novel video. This is achieved by estimating the corresponding grouping criteria Γ_i using the regression learned from labeled video scene examples from other videos in the training set.

6. Inferring Social Communities

Now that we have formulated grouping criteria using low-level audiovisual features in film scenes, we turn next to inferring communities in social network graph data structure. With regards to actors co-occurring in a scene, we hold the view that frequency of co-occurrence is correlated to similar membership. That is, actors of the same community co-occur more frequently. The reasoning behind our conjecture is that pair of actors with a friendly social relation will occur in friendly scenes. But they will also co-occur in adversarial scenes which involve conflict between multiple people. On the other hand a pair of actors that have an adversarial social relation will only co-occur in exclusively adversarial scenes. This is reflected in the grouping criteria wherein, higher the number of co-occurrences for community members, higher the value of the positive grouping criteria compared to the negative grouping criteria.

6.1. Social Network Graph

We denote the occurrence of an actor c_i in a video by a boolean appearance function $\Lambda_i : T \rightarrow \{0,1\}$, based on time where the duration of the video is $T \subset \mathbb{R}^+$. Naturally, in implementation we only have access to its sampled version. Let the sampling period be of length t seconds. We satisfy Nyquist sampling theorem. Accordingly, as long as $t \leq \min_i\{1/2B_i\}$, where B_i is the highest frequency of the actor i's appearance function, information regarding both the continuous appearance and the actor's co-occurrences can be determined from those discrete samples.

In our formulation a video $\mathbb{V}$ is considered to constitute on-overlapping M scenes, where each scene s_i contains social interactions among actors occurring in the same scene. We approximate the appearance functions of actors as a scene-actor relation matrix denoted by $A = \{A_{i,j}\}$, where $A_{i,j} = 1$ if there exists $t \in L_i$, where L_i is the time interval of s_i, such that $\Lambda_j(t) = 1$. For a feature film this can be obtained by searching for mention of corresponding actor names in the film script. This representation is reminiscent of the actor-event graph in social network analysis. Although actor relations in A can directly be applied to construct a social network, we shall quantitatively show that utilization of audiovisual features leads to a better social network representation. This should also be intuitively obvious since audiovisual features give us a real number valued measure of the degree of affinity between different actors, whereas simple co-occurrence is a binary valued feature.

The actor social network is represented as an undirected graph $G(V,E)$ with cardinality $|V|$. In this graph, the nodes represent the actors

$$V = \{v_i : \text{node } v_i \sim \text{actor } c_i\} \tag{2}$$

and the edges define the interactions between the actors

$$E = \{(v_i, v_j) | v_i, v_j \in V\} \tag{3}$$

The graph G is fully-connected with an affinity matrix K of size $|V| \times |V|$, since any two actors can potentially co-occur in any scene. The element in the affinity matrix $K(c_i, c_j)$ for two actors c_i and c_j is a real-valued score, which is decided by an affinity learning method. The values in the affinity matrix serve as the basis for social network analysis. This includes estimating social communities and also the leader in each of these estimated communities.

6.2. Actor Interaction Model

Let c_i be actor indexed i, and $\mathbf{f} = (f_1, \cdots, f_N)^T$ be the vector of community memberships containing ± 1 values, where f_i refers to the membership of c_i. Let **f** distribute according to a zero-mean identity-covariance Gaussian process

$$P(\mathbf{f}) = (2\pi)^{-N/2} \exp^{-\frac{1}{2}\mathbf{f}^T\mathbf{f}} \tag{4}$$

In order to model the information contained in the scene-actor relation matrix A and the aforementioned grouping criteria of each scene Γ_i, we assume the following distributions:

1. if actors c_i and c_j co-occur in a friendly scene k ($\Gamma_k \geq 0$), then $f_i - f_j \sim \mathcal{N}(0, \frac{1}{\Gamma_k^2})$
2. if actors c_i and c_j co-occur in an adversarial scene k ($\Gamma_k < 0$), then $f_i + f_j \sim \mathcal{N}(0, \frac{1}{\Gamma_k^2})$

So, if $\Gamma_i = 0$, then the constraint imposed by a scene is rendered inconsequential. This corresponds to the least confidence in the constraint. On the other hand, if $\Gamma_i = \pm 1$, the corresponding constraint becomes the strongest. Due of the nature of distributions we use, none of the constraints are hard, which makes our model robust to prediction errors. Applying the Bayes' rule, the posterior probability of **f** given the constraints is defined in a continuous formulation as the following:

$$\begin{aligned} P(\mathbf{f}|\{\Lambda_k\},\Gamma) = P(\mathbf{f})\exp\{-\sum_{i,j}\int_{t\in\{t:\Gamma(t)\geq 0\}} \Lambda_i(t)\Lambda_j(t)\frac{(f_i - f_j)^2\Gamma(t)^2}{2}dt \\ -\sum_{i,j}\int_{t\in\{t:\Gamma(t)<0\}} \Lambda_i(t)\Lambda_j(t)\frac{(f_i + f_j)^2\Gamma(t)^2}{2}dt\} \\ \propto \exp\{-\frac{1}{2}\mathbf{f}^T\mathbf{f} - \sum_{i,j}\int_{t\in\{t:\Gamma(t)\geq 0\}} \Lambda_i(t)\Lambda_j(t)\frac{(f_i - f_j)^2\Gamma(t)^2}{2}dt \\ -\sum_{i,j}\int_{t\in\{t:\Gamma(t)<0\}} \Lambda_i(t)\Lambda_j(t)\frac{(f_i + f_j)^2\Gamma(t)^2}{2}dt\}. \end{aligned} \tag{5}$$

Translating this equation into its discrete version, we have:

$$\begin{aligned} P(\mathbf{f}|A,\Gamma) &= P(\mathbf{f})\Pi_{k:\Gamma_k\geq 0}\Pi_{c_i,c_j\in s_k}\exp\frac{-(f_i-f_j)^2\Gamma_k^2}{2} \\ &\quad \Pi_{k:\Gamma_k<0}\Pi_{c_i,c_j\in s_k}\exp\frac{-(f_i+f_j)^2\Gamma_k^2}{2} \\ &\propto \exp\{-\frac{1}{2}\mathbf{f}^T\mathbf{f}-\sum_{k:\Gamma_k\geq 0}\sum_{c_i,c_j\in s_k}\frac{(f_i-f_j)^2\Gamma_k^2}{2} \\ &\quad -\sum_{k:\Gamma_k<0}\sum_{c_i,c_j\in s_k}\frac{(f_i+f_j)^2\Gamma_k^2}{2}\}. \end{aligned} \tag{6}$$

It can be verified that $P(\mathbf{f}|A,\Gamma) \propto \exp(-\frac{1}{2}\mathbf{f}^T K^{-1}\mathbf{f})$ is a 0 mean Gaussian Process. Using $K_{i,j} = E\{f_i f_j|A,\Gamma\}$ as the learned affinity between c_i and c_j, it follows that $K = M^{-1}$, where

$$M_{i,j} = \left\{ \begin{array}{ll} \sum_{k:c_i,c_j\in s_k,\Gamma_k<0}\Gamma_k^2 - \sum_{k:c_i,c_j\in s_k,\Gamma_k\geq 0}\Gamma_k^2 & \text{if } i\neq j \\ 1+\sum_{l\neq i}\sum_{k:c_i,c_l\in s_k}\Gamma_k^2 & \text{if } i=j \end{array} \right. . \tag{7}$$

The resulting matrix, K, is symmetric and positive definite. However, unlike an affinity matrix from a Gaussian kernel, it may contain negative values. Our approach has two special cases. In the first case where $\Gamma_i = 1$, the aforementioned learning mechanism reduces to a co-occurrence based approach which is a traditional tool in social network analysis [11, 17]. Specifically, $M_{i,j}$, for $i \neq j$, represents the minus value of the number of scenes where c_i and c_j occur together. This reduced scheme does not utilize the audiovisual feature based prediction of grouping criteria, and serves as a natural baseline in this article. In the second case, if we use fixed variance parameters in the assumed distributions instead of the learned ones, our affinity learning method reduces to the affinity propagation approach proposed in [30].

7. Social Network Analysis

The principal objectives of social network analysis in films is discovering grouping of actors to that comprise social communities and finding the most influential actor within each such community. Typically, communities are detected using spectral clustering techniques for sociological data. An example is the commonly used modularity-cut algorithm [11]. The authors in [17] have shown that the performance of the modularity cut algorithm can be increased by introducing a generalized objective referred to as the max-min modularity. This max-min modularity clustering, however, ignores edge attributes. As such it can not be simply applied to our social network of actors, which does have weighted edges between actors.

7.1. Assignment to Communities

We first compute a principal affinity matrix K' with the condition: $K'_{i,j} = K_{i,j}$ for $K_{i,j} > 0$, and $K'_{i,j} = 0$ for other entries. Next we compute a complementary affinity matrix K'' with the condition: $K''_{i,j} = -K_{i,j}$ for $K_{i,j} < 0$, and $K''_{i,j} = 0$ for other entries. The matrix K'' represents the degree of lack of relation between actors in the graph in terms of community memberships. Adopting the strategy in [17] and using K' and K'', we formulate the max-min modularity criterion as $Q_{MM} = Q_{max} - Q_{min}$ for:

$$Q_{max} = \frac{1}{2m'}\sum_{i,j}(K'_{ij} - \frac{k'_i k'_j}{2m'})(f_i f_j + 1) \triangleq \frac{1}{2m'}\sum_{i,j} B'_{i,j}(f_i f_j + 1), \tag{8}$$

$$Q_{min} = \frac{1}{2m''}\sum_{i,j}(K''_{ij} - \frac{k''_i k''_j}{2m''})(f_i f_j + 1) \triangleq \frac{1}{2m''}\sum_{i,j} B''_{i,j}(f_i f_j + 1), \tag{9}$$

where $m' = \frac{1}{2}\sum_{ij} K'_{ij}$, $k'_i = \sum_j K'_{ij}$, $m'' = \frac{1}{2}\sum_{ij} K''_{ij}$, $k''_i = \sum_j K''_{ij}$ and the term $\frac{k'_i k'_j}{2m'}$ represents the expected edge strength between the actors c_i and c_j [11]. Based on this observation, we note that $K'_{i,j} - \frac{k'_i k'_j}{2m}$ measures how much the connection between two actors is stronger than what would be expected between them, and serves as the basis for keeping the two actors in the same community. In our formulation, the max-min modularity Q_{MM} arises from the conditions for a good network division, such that: connectivity weight between communities should be smaller than expected; and assignment of actors not friendly to each other to the same community should be minimized. These conditions can be realized by maximizing Q_{MM}. Using standard eigenvector analysis, the eigenvector $\mathbf{u}$ of $\frac{1}{2m'}B' - \frac{1}{2m''}B''$ with the largest eigenvalue maximizes a relaxed version of Q_{MM}. The resulting eigenvector solution contains real values, and we threshold them at the 0 level to obtain the desired community assignments for the actors. We let $f_i = +1$ if $u_i \geq 0$, and $f_i = -1$ if otherwise.

7.2. Estimating Community Leader

Subsequent to assignment of actors to communities, we next estimate the leaders of each community. We do this by analyzing the centrality of each actor in the

Films	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
# of scenes	198	151	238	226	116	51	105	297	188	199
# of captions in lines	1143	1585	1063	1155	1337	1515	1293	1262	1099	1402
# of actors in total	11	7	10	10	7	7	6	8	10	9
# of actors in community 1	6	4	6	5	3	3	4	6	7	7

Table 2: Statistics of movies in our dataset which includes the number of scenes in the film, the number of lines in closed caption data, the total number of actors in the movie and the number of actors in one of the two communities.

community. The literature in sociology defines the centrality of a member in a social network by its degree of betweenness [31]. Instead of this definition of centrality, we adopt a new measure which we refer to as the Eigen-centrality [32]. We do this since it aligns well with out approach to community estimation. Let the centrality score, x_i for the i^{th} actor be proportional to the sum of the scores of all actors which are connected to it: $x_i = \frac{1}{\lambda}\sum_{j=1}^{N} K'_{i,j} x_j$, where N is the total number of actors in the video and λ is a constant. It follows from this notation that the centralities of actors satisfy $K'\mathbf{x} = \lambda\mathbf{x}$ in the vector form. It can be shown that the eigenvector with largest eigenvalue provides the desired centrality measure [32]. Therefore, if we let the eigenvector of K' with the largest eigenvalue be $\mathbf{v}$, the leaders of the two communities are given by $\arg\max_{i:u_i \geq 0} v_i$ and $\arg\max_{i:u_i < 0} v_i$ respectively. In our problem domain, when the communities correspond to two adversarial social groups in films, their respective leaders frequently are the protagonist and the antagonist in the feature film.

8. Experiments

In this section we describe our dataset creation approach and evaluation of methodology for estimating two communities in each film along with the leader of each community.

8.1. The Dataset

In our experiments we create a dataset of 10 feature films. These films span multiple genres including action, adventure, fantasy and drama. We chose to include the following films: (1) *G.I. Joe: The Rise of Cobra (2009)*; (2) *Harry Potter and the Half-Blood Prince (2009)*; (3) *Public Enemies (2009)*; (4) *Troy (2004)*; (5) *Braveheart (1995)*; (6) *Year One (2009)*; (7) *Coraline (2009)*; (8) *True Lies (1994)*; (9) *The Chronicles of Narnia: The Lion, the Witch and the Wardrobe*

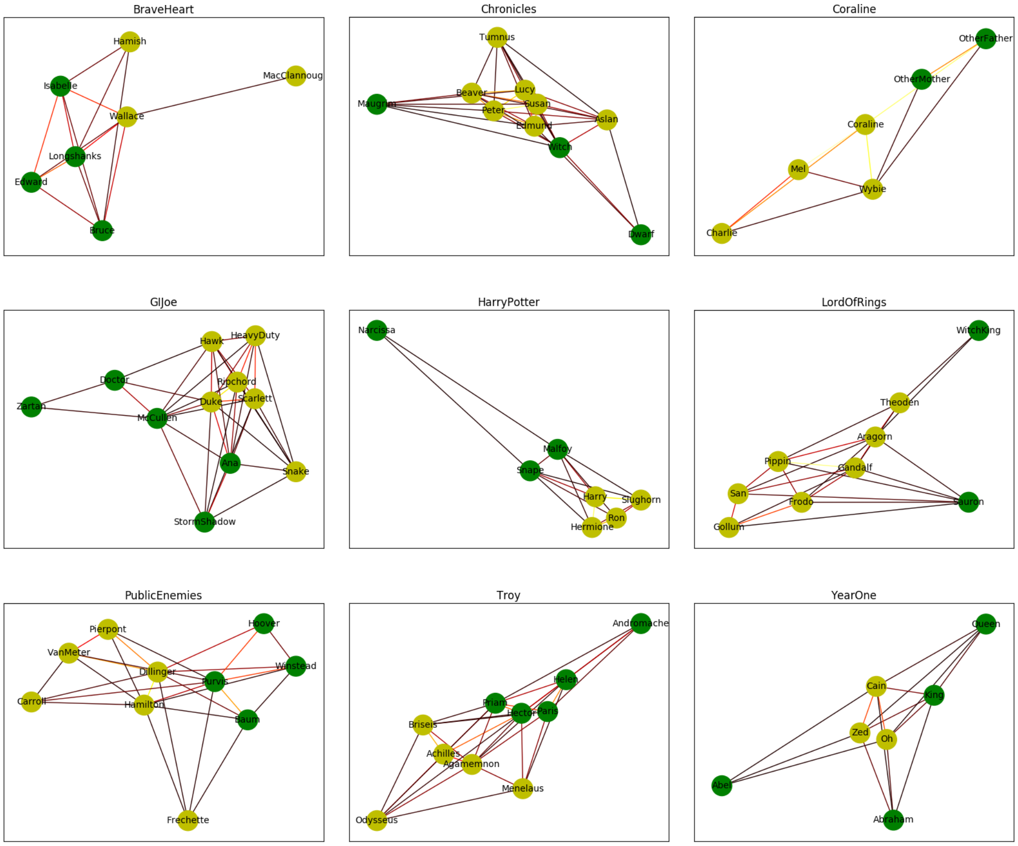


Figure 10: Social networks generated using the proposed approach for the films in our dataset. Actors (nodes) are colored according to their community membership. The strength of affinity is indicated by the edge attribute, where the stronger the edge is the warmer the edge color.

(2005); (10) *The Lord of the Rings: The Return of the King (2003)*. These films have two discernible communities that have a adversarial social relationship with each other. Each of the communities also has a discernible leader. In some films, the community leader is obvious, whereas in some films there are communities with multiple actors of comparable influence. We compiled and processed the video file and other meta-information files for each film. Relevant statistics for each film is detailed in Table 2. This dataset contains visual and auditory features, film script, and closed caption data. These features and meta-data are temporally aligned.

8.2. Audio-Visual Alignment

We segment each feature film into scenes to aid computing our actor-scene occurrence matrix. We also need to align visual and auditory features in the video. We turn to the film script to help us do this task. The script also provides us a start and stop time stamp for scenes in the entire film. The script itself is typically available in the public domain in its draft version with no time tag information. It also has no professional editing information. The closed captions are composed of lines d_i, which contain actor dialogues. Our method is a variant of the alignment technique in [33]. We first divide the script into scenes, each of which is denoted as s_i. Similarly, closed captions are divided into lines d_i. Next we define $\mathscr{C}$ to be a cost matrix. We compute the percentage p of words in the closed caption d_j matched with scene s_i, while respecting the order of words. We set the cost as $\mathscr{C}_{i,j} = 1 - p$.

Finally, we apply dynamic time warping to $\mathscr{C}$ to estimate the start t_1^i and stop times t_2^i of s_i. These correspond to the smallest and largest time stamps for closed captions matched with scene s_i. After processing the entire film we have the start and stop times of every scene in the film. Of course scripts in public domain are not accurate and therefore the consequent scene segmentation will not be perfect. However, potential errors community and leader estimation and due to this small flaw in alignment is very small and can be ignored in practice. Unless the alignment error frequently causes different actors to occur in different scenes than intended our approach is robust to such inaccuracies.

8.3. Social Affinity

We evaluate our social networks with their accompanying affinity matrices. We experiment for two scenarios. In the first case the co-occurrence data is acquired in the matrix A from the script. This is the standard approach in sociology. In the second case, we use the matrix A in conjunction with scene-level grouping criteria Γ_i, which we have learned using our audiovisual features. Since we want to quantitatively evaluate the degree of contribution of the audiovisual features, we provide comparisons of collective use of visual and auditory features with their individual use in estimation of communities and their leaders. In Figure 10, we show a graphical representations of the social networks graphs for the 10 films in our dataset learned from both visual and auditory features. Supporting our approach the connectivity between inter-community actors is typically weaker than connectivity between intra-community actors.

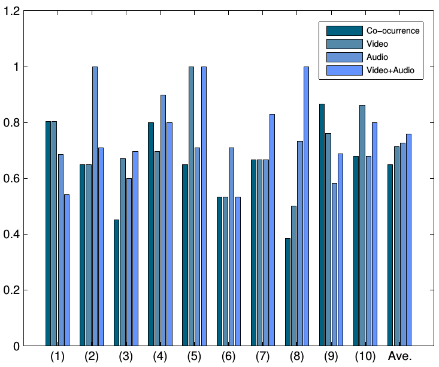


Figure 11: Performance analysis of Social Network using F_1 score. We compare relative performance using co-occurrence information, just the video and just the audio and finally a combined audio-visual feature. The result demonstrate that our use of audio-visual features for creating the social network provides the best overall performance.

8.4. Community Assignment

The affinity between the actors is strongly related to the grouping criteria of the scenes in which the actors occur. This result serves as validation of our choice of SVR in its efficacy at computing the grouping criteria. Accordingly, for our regression approach we calculate the error over all scenes in each of the 10 films. We use the Mean Square Error (MSE) measure. The expectation of the cumulative MSE using both visual and auditory features is 0.61. When each of the content features is utilized separately the expected MSE is higher. For only visual features the expected MSE is 0.80 and for only auditory features the expected MSE is 0.77. This result validates are conjecture regarding the significance of audio-visual data towards estimating social relationships.

We also computed the accuracy of our classification, where we predict a scene as being predominantly friendly or adversarial in terms of the interaction between the actors in that scene. Again we performed our classification using features that were audiovisual, only visual, and only auditory. We computed prediction accuracy of 0.816, 0.782, and 0.787 respectively. These results indicate that our estimate of grouping criteria can be effectively employed to draw inference regarding the social relation between actors in the film.

These result on prediction accuracy correspond to precision in our assignment of actors in each film into either of the protagonist or antagonist communities. Since a social community is a group of actors, our prediction efficacy can be measured using the precision and recall values of actor assignments in comparison to the ground truth. For both communities these two measures can be combined into an F_1 score. The F_1 score is the harmonic mean of precision and recall. It is a widely adopted prediction or retrieval metric. The significance of the F_1 score is that it does not require the cardinality of groups with different class labels to be approximately equal to be reliable. Since the number of members of our two protagonist and antagonist communities can be considerably different, we chose to use the F_1 score. We compute the average F_1 score of our two communities, that comprises the final detection accuracy for each film. We computed the F_1

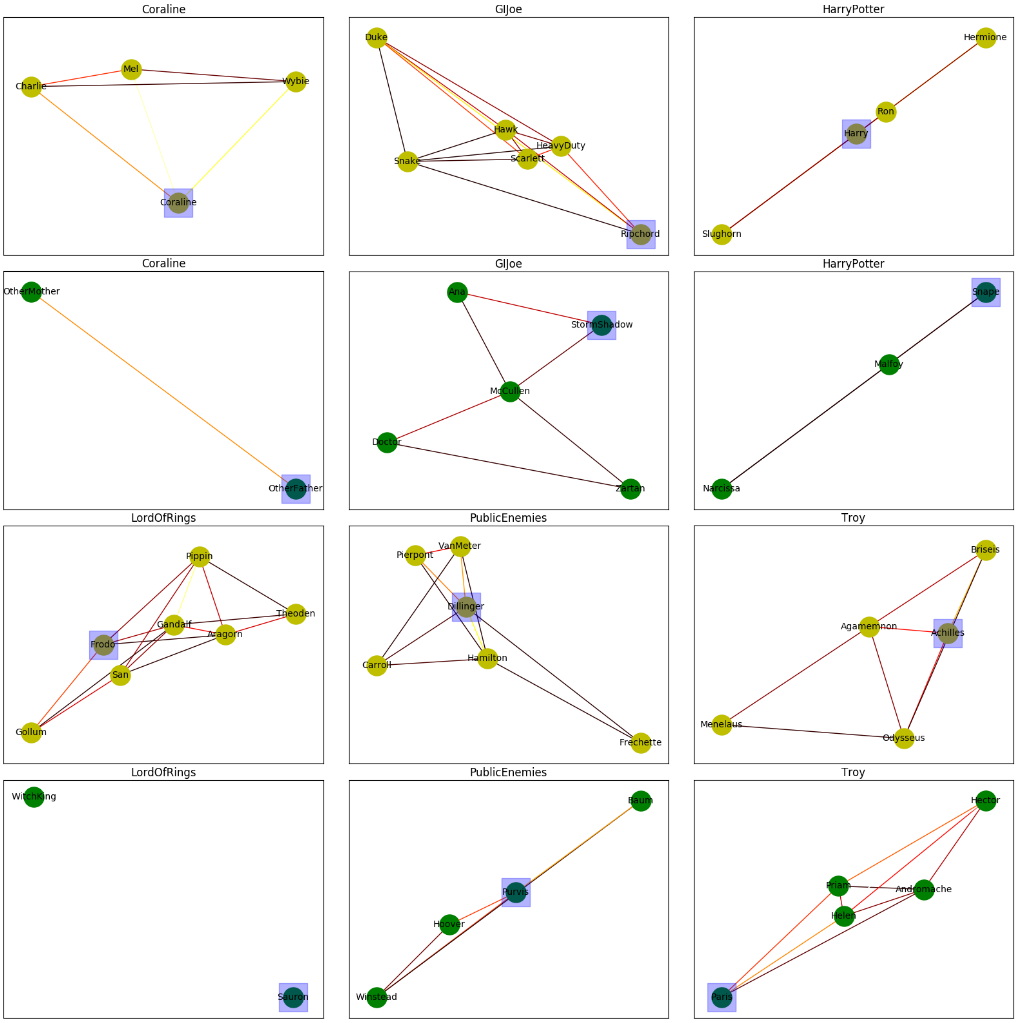


Figure 12: Estimated community leaders using our approach for each of the two communities in each of the films.

Movies	(1)	(2)	(3)	(4)	(5)
Community 1	*Ripchord*	**Harry**	**Dillinger**	**Achilles**	**Wallis**
Community 2	*StormShadow*	**Snape**	**Purvis**	**Paris**	**Longshanks**
Movies	(6)	(7)	(8)	(9)	(10)
Community 1	**Oh**	**Coraline**	**Harry**	*Susan*	**Frodo**
Community 2	*Abraham*	**Other Father**	**Salim**	**Witch**	**Sauron**

Table 3: Community leaders discovered using the proposed framework. The names in bold face refer to correct ones, whereas those in italics are not.

score of our predictions for each of the ten films. We report these results when using co-occurrence information, visual, audio, and audiovisual features. The results are reported in Figure 11. Supporting our conjecture, prediction performance improves with utilization of more informative features. We typically get the lowest F_1 scores using only the co-occurrence information and the highest scores when using both auditory and visual features. We also observe that when using independent features, the auditory features have a small performance edge over visual features. The combined audiovisual features of course have the best performance, and the expectation of their prediction F_1 score is 76.0%. To put this result in proper perspective, the use of audiovisual features when compared to exclusive use of co-occurrence information improves the assignment performance by 11.1%.

8.5. Actor Affinity

Following from our earlier discussion in Section 7 on the assignment of actors to communities, which we achieved by the analysis of the Eigen-space of $\frac{1}{2m'}B' - \frac{1}{2m''}B''$. The community leaders computed using Eigen-analysis are shows in Figure 12. It is also evident from the figure that, typically, the actors who are associated with different communities tend to be mapped further apart.

8.6. Community Leaders

The eigenvector of K' with the highest eigenvalue is expected to be the leader in each of the social communities. In Figure 12, we display the estimated leaders using this approach. For ease of interpreting the results, we show a picture of the estimated leader in each of the two adversarial communities for each film. The ground truth of community leaders is: (1) Duke and McCullen; (2) Harry and Snape; (3) Dillinger and Purvis; (4) Achilles and Paris; (5) Wallace and Longshanks; (6) Oh and King; (7) Coraline and Other Father; (8) Harry and Salim; (9)

Aslan and Witch; (10) Frodo and Sauron. When our estimated leaders correspond to the actual leaders in the film, we indicate it by the actors name in bold face, while wrongly estimated leader's names are shown in italics. More often than now, we are able to successfully predict the actual leader in both the adversarial communities.

9. Latent Features

We have now established our approach as an effective way to discover adversarial communities in films, assign actors to the two communities, and finally predict the leader of each community. We began with actor co-occurrence and built upon that with the use of low-level audiovisual features in scenes and demonstrated a consistent improvement in performance. Now we discuss a further improvement to our existing framework. Recall that the underlying idea behind the improvement in community assignment from audiovisual features was that social expression is contextual. In other words, there is a bi-jective relation between the nature of content in the scenes and the nature of the relationship between actors in that scene. We move further from low-level features to the use of latent features. A latent feature is by definition not directly observable, but computed using low-level features. It typically have greater semantic relevance than low-level features. Our conjecture in this article is that the semantic meaning of a scene is correlated to the predominant social activity in that scene. Our latent features correspond to scenes with semantic meaning that include social activity like 'shooting', 'fighting', 'running', etc.

The intuition is that actors co-occurring in a scene that is a 'fighting' scene tend to be adversaries, whereas actors co-occurring in a 'talking' scene tend to be friendly. Since the semantic meaning is derived from visual latent features, we also refer to it as visual concept. We employ the set of visual concepts provided in [34]. This data is available in public domain and is sufficiently broad in its scope to be useful in out application. It has 374 trained SVM classifiers, each for a different semantically relevant latent feature.

To compute the latent feature we use 3 low-level features according to [34], since the classifiers are also trained on these features. These include color, texture and edges in images, where each image is a key-frame in the film. The color feature is based on the LUV color space. It is called Grid Color Moment (GCM). We convert the video frame from RGB to LUV color space. We then compute the first 3 moments of the 3 channels over a 5×5 sized window in the entire frame. We aggregate this feature into a single 225-dimensional feature vector.

The texture feature is computed using the Gabor filter. In our implementation the Gabor filter has 4 scales and 6 orientations. We utilize the mean and standard deviation acquired from running the filter on the video frame. This results in a texture feature in the form of a vector with 48 dimensions. The edge content in the image is modeled using the Edge Direction Histogram (EDH), which is composed of 73 dimensions corresponding to 72 bins of edge direction quantized at 5 degree intervals and 1 bin for non-edge points.

Note that we are computing features at the scene level rather than individual frames. First we use the start and stop time stamps to ascertain the set of video frames that pertain to a scene under consideration. In our framework, the key-frames in a video are representative of that video. After the key-frames for a scene are selected, we compute low-level features for each of these key-frames. We train a logistic SVM for these features, and we train a separate SVM for each feature. Then we combine the output of the SVMs. For the i^{th} key-frame, and our three features indexed $j = \{1,2,3\}$, the SVM output is $f_{i,j}$. The average of the three scores $f_i = \frac{1}{3}(f_{i,1} + f_{i,2} + f_{i,3})$ is used as the overall score for the i^{th} key-frame. Once we have computed a score for each key-frame, we use max-pooling over all key-frames in the scene to compute a latent feature value of the scene given as $\max_i \overline{f}_i$. We follow this scheme for all 374 visual concepts in [34], which gives us a 374-dimensional latent feature vector for a scene.

Each dimension of the latent feature vector provides a confidence score corresponding to one semantic concept. Different dimensions of the latent vector carry different levels of relevant information, specifically towards the task of inferring social activity in the scene. In addition, some visual concepts may not receive sufficient data in the video or typically show large variability to be reliable for an inference mechanism. To resolve such problems, we use a supervised dimension reduction method to prune out undesirable dimensions. We use Kernel Local Fisher Discriminant Analysis (KLFDA) [35]. This is a transformation that embedded the latent feature vector from its original dimensions to a lower dimensional feature space. The embedded representation of the latent feature is more informative and compact.

As stated previously in Section 5, in our framework a video is composed of M scenes, $s_1, s_2 \cdots s_M$, each of which contains a set of actors and has an associated grouping criteria Γ_i. The grouping criteria assigns actors to communities and thereby determines if a pair of actors that appear in the same scene have a friendly or adversarial social relation. To estimate the grouping cues Γ_i from the d-dimensional embedded latent vectors, we use SVR with a RBF kernel. We analysis the performance of community prediction accuracy using latent features

Methods	Prec(+)	Prec(-)	Prec(ave)	F_1
Observed features	–	–	78.2%	76.0%
Latent features (d=50)	83.1%	85.0%	84.1%	81.9%

Table 4: A comparative analysis of features for learning social network. The measures used are as follows: Prec(+): precision of Γ estimates for the positive class; Prec(-): precision of Γ estimates for the negative class; Prec(ave): average precision; and F_1 measure for community assignment prediction averaged over all films.

on our dataset of 10 films. A comparison of its performance relative to our basic framework is reported in Table 4. We note that the performance improvement conferred by use of latent features is partially accounted for by the community detection method discussed in [1]. Nevertheless, it is made clear from our work that the use of latent features describing the semantically relevant content in scenes provides better grouping criteria than using only low-level features.

10. Summary

We set out to acquire information from video that could be used to group people in those videos into communities based on the sociologically relevant activities of these people. Towards this objective we argued that visual and auditory content in video has a correlation with the social interaction. Consequently, we can utilize audiovisual features to predict social community membership of people who co-occur in a scene. We build a basic framework that used low-level visual and auditory features. We chose entropy based descriptors which provide an aggregated measure of the degree of activity in a scene. We worked with the intuition that scenes with greater activity in both visual and auditory features correspond to dramatic situations which typically happen when adversaries meet. Scenes containing members from the same social group have comparatively lesser activity or that activity, defined by motion field flow is in harmony or uni-directional. We constructed quantitative descriptors for these intuitions into a grouping criteria. We used a set of ten feature films to evaluate our framework. The precision-recall score of our community membership predictions demonstrate that our framework is effective at modeling the social network that exists among the people in the films.

We extended this basic framework by using latent features. Again, we proceeded with the intuition that friendly or adversarial scenes had a correlation with the semantics of the activity in the scene. We created a measure of visual concept in the scene using latent features. This additional semantically relevant information helped further improve our community membership prediction accuracy.

Although we have used feature films to validate our framework, it can be applied to other types of videos as well. Future work would involve extending our framework to diverse application domains like anthropology and zoology. Analysis of video of birds and animals could be used to identify social structures and groups which would help improve understanding of those species in their respective fields. We will consider automated analysis of videos uploaded to social network websites like ‘YouTube’. These videos can be summarized from the perspective of ‘social expression content’, which in turn can be used to improve video retrieval, where a query can now include the desired genre like action or drama of retrieved videos.

References

[1] L. Ding, A. Yilmaz, Inferring social relations from visual concepts, Proc. IEEE Int. Conf. Comput. Vis. (2011) 699–706doi:10.1109/ICCV.2011.6126306.

[2] I. Laptev, On space-time interest points, Int. J. Comput. Vis. 64 (2-3) (2005) 107–123. doi:10.1007/s11263-005-1838-7.

[3] A. A. Efros, A. C. Berg, G. Mori, J. Malik, Recognizing Action at a Distance, in: Int. Conf. Comput. Vis., IEEE, Nice, France, 2003, pp. 1–8.

[4] S. Ali, A. Basharta, M. Shah, Chaotic Invariants for Human Action Recognition, in: Int. Conf. Comput. Vis., IEEE, 2007, pp. 1–8.

[5] A. Yilmaz, M. Shah, A differential geometric approach to representing the human actions, Comput. Vis. Image Underst. 109 (3) (2008) 335–351. doi:10.1016/j.cviu.2007.09.006.

[6] N. Kyriazis, A. Argyros, Physically plausible 3D scene tracking: The single actor hypothesis, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (2013) 9–16doi:10.1109/CVPR.2013.9.

[7] A. Fathi, G. Mori, Action recognition by mid-level motion features, IEEE Int. Conf. Comput. Vis. Pattern Recognit.

[8] J. Alon, V. Athitsos, Q. Yuan, S. Sclaroff, A unified framework for gesture recognition and spatiotemporal gesture segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (9) (2009) 1685–1699. doi:10.1109/TPAMI.2008.203.

[9] A. Yilmaz, M. Shah, Recognizing human actions in videos acquired by uncalibrated moving cameras, Proc. IEEE Int. Conf. Comput. Vis. I (2005) 150–157. doi:10.1109/ICCV.2005.201.

[10] Y. Song, L.-p. Morency, R. Davis, Action Recognition by Hierarchical Sequence Summarization, in: Comput. Vis. Pattern Recognit., IEEE, 2013. doi:10.1109/CVPR.2013.457.

[11] M. Newman, Modularity and community structure in networks, Pnas 103 (23) (2006) 8577–8582. arXiv:0602124v1, doi:10.1073/pnas.0601602103.

[12] T. Yang, Y. Chi, S. Zhu, Y. Gong, R. Jin, A Bayesian Approach Toward Finding Communities and Their Evolutions in Dynamic Social Networks, Proc. SIAM Int. Conf. Data Min. (2009) 990–1001arXiv:arXiv:0906.0612v2, doi:10.1088/1742-5468/2008/10/P10008.

[13] J. Q. J. Qiu, Z. L. Z. Lin, C. T. C. Tang, S. Q. S. Qiao, Discovering Organizational Structure in Dynamic Social Network, 2009 Ninth IEEE Int. Conf. Data Min. (2009) 932–937doi:10.1109/ICDM.2009.86.

[14] T. Yu, S. N. Lim, K. Patwardhan, N. Krahnstoever, Monitoring, recognizing and discovering social networks, 2009 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. CVPR Work. 2009 (2009) 1462–1469doi:10.1109/CVPRW.2009.5206526.

[15] Z. Yun, S. Mubarak, Video Scene Segmentation Using Markov Chain Monte Carlo, Multimedia, IEEE Trans. 8 (4) (2006) 686–697. doi:10.1109/TMM.2006.876299.

[16] O. Arandjelovic, A. Zisserman, Automatic Face Recognition for Film Characters Retrieval in Feature-Length Films, Proc. 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.

[17] J. Chen, O. R. Zaïane, R. Goebel, Detecting Communities in Social Networks using Max-Min Modularity, From Sociol. to Comput. Soc. Networks (2010) 197–214.

[18] C.-Y. Weng, W.-T. Chu, Ja-Ling Wu, RoleNet: Movie Analysis from the Perspective of Social Networks, Multimedia, IEEE Trans. 11 (2) (2009) 256–271. doi:10.1109/TMM.2008.2009684.

[19] W. Ge, R. T. Collins, B. Ruback, Automatically detecting the small group structure of a crowd, 2009 Work. Appl. Comput. Vision, WACV 2009doi:10.1109/WACV.2009.5403123.

[20] A. Fathi, J. K. Hodgins, J. M. Rehg, Social interactions: A first-person perspective, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (2012) 1226–1233doi:10.1109/CVPR.2012.6247805.

[21] V. Ramanathan, B. Yao, L. Fei-Fei, Social role discovery in human events, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (2013) 2475–2482doi:10.1109/CVPR.2013.320.

[22] Y. Fan, C. R. Shelton, Learning continuous-time social network dynamics, in: Proc. Twenty-Fifth Conf. Uncertain. Artif. Intell., 2009, pp. 161–168. arXiv:1205.2648.

[23] N. Eagle, A. S. Pentland, D. Lazer, Inferring friendship network structure by using mobile phone data., Proc. Natl. Acad. Sci. U. S. A. 106 (36) (2009) 15274–15278. doi:10.1073/pnas.0900282106.

[24] N. Eagle, A. S. Pentland, Eigenbehaviors: Identifying structure in routine, Behav. Ecol. Sociobiol. 63 (7) (2009) 1057–1066. doi:10.1007/s00265-009-0739-0.

[25] D. Zhang, D. Gatica-Perez, S. Bengio, I. McCowan, Modeling individual and group actions in meetings with layered HMMs, IEEE Trans. Multimed. 8 (3) (2006) 509–520. doi:10.1109/TMM.2006.870735.

[26] M. Al-Hames, C. Lenz, S. Reiter, J. Schenk, F. Wallhoff, G. Rigoll, Robust multi-modal group action recognition in meetings from disturbed videos with the asynchronous Hidden Markov model, Proc. - Int. Conf. Image Process. ICIP 2 (2007) 213–216. doi:10.1109/ICIP.2007.4379130.

[27] Z. Rasheed, M. Shah, Movie genre classification by exploiting audio-visual features of previews, Object Recognit. Support. by user Interact. Serv. Robot. 2 (i) (2002) 1086–1089.

[28] A. Yilmaz, O. Javed, M. Shah, Object tracking: A Survey, ACM Comput. Surv. 38 (4) (2006) 1–44.

[29] J. Lin, W. Wang, Weakly-supervised violence detection in movies with audio and video based co-training, Lect. Notes Comput. Sci. 5879 LNCS (2009) 930–935.

[30] Z. Lu, M. A, Constrained Spectral Clustering through Af nity Propagation, Comp. A J. Comp. Educ. (M) (2008) 1–8. doi:10.1109/CVPR.2008.4587451.

[31] L. C. Freeman, Centrality in social networks conceptual clarification, Soc. Networks 1 (3) (1978) 215–239. doi:10.1016/0378-8733(78)90021-7.

[32] A. Landherr, B. Friedl, J. Heidemann, A critical review of centrality measures in social networks, Bus. Inf. Syst. Eng. 2 (6) (2010) 371–385.

[33] T. Cour, C. Jordan, E. Miltsakaki, B. Taskar, Movie/Script: Alignment and Parsing of Video and Text Transcription, in: Eur. Conf. Comput. Vis., Vol. 9905, 2008, pp. 158–171. arXiv:1311.2901, doi:10.1007/978-3-319-46448-0.
URL http://link.springer.com/10.1007/978-3-319-46448-0

[34] A. Yanagawa, S.-f. Chang, L. Kennedy, W. Hsu, Columbia University ' s Baseline Detectors for 374 LSCOM Semantic Visual Concepts, Tech. rep., Columbia University (2007).

[35] M. Sugiyama, Dimensionality Reduction of Multimodal Labeled Data by Local Fisher Discriminant Analysis, J. Mach. Learn. Res. 8 (2007) 1027–1061.