Social-Event-Driven Camera Control for Multicharacter Animations

I-Cheng Yeh, Wen-Chieh Lin, *Member*, *IEEE*, Tong-Yee Lee, *Senior Member*, *IEEE*,
Hsin-Ju Han, Jehee Lee, and Manmyung Kim

Abstract—In a virtual world, a group of virtual characters can interact with each other, and these characters may leave a group to join another. The interaction among individuals and groups often produces interesting events in a sequence of animation. The goal of this paper is to discover social events involving mutual interactions or group activities in multicharacter animations and automatically plan a smooth camera motion to view interesting events suggested by our system or relevant events specified by a user. Inspired by sociology studies, we borrow the knowledge in Proxemics, social force, and social network analysis to model the dynamic relation among social events and the relation among the participants within each event. By analyzing the variation of relation strength among participants and spatiotemporal correlation among events, we discover salient social events in a motion clip and generate an overview video of these events with smooth camera motion using a simulated annealing optimization method. We tested our approach on different motions performed by multiple characters. Our user study shows that our results are preferred in 66.19 percent of the comparisons with those by the camera control approach without event analysis and are comparable (51.79 percent) to professional results by an artist.

Index Terms—MOCAP, multicharacter animation, event analysis, social network analysis.

1 Introduction

Motion capture data have emerged as important resource for making computer animation and digital special effects in recent years. As there are more and more motion capture data, automatic camera control for discovering and viewing interesting events in motion capture data can be very useful for many applications such as 3D authoring tools, motion capture data preview, game recording and replaying functions. In computer games like massively multiplayer online role-playing games (MMORPGs) and second life, which allow many users to interact with each other in virtual worlds, the users of such a virtual world often want to know about interesting events happening or happened in the virtual world. These events are sometimes planned/scripted, but sometimes happen spontaneously as a result of interaction among the users. An automatic camera control approach that respects interaction events will help the users to browse or understand the activities occurring in the virtual world. As mentioned in [6], a summary of a past gameplay can help users maintain engaged in their games.

In a multiple-character animation, a group of characters may interact with each other, e.g., passing objects or moving objects collaboratively. The interaction among individuals and groups often produces interesting events. In this paper, our goal is to identify social events involving mutual interactions or group activities in multicharacter animations and automatically plan a smooth camera motion to view interesting events suggested by our system or relevant events specified by a user. Although previous work has addressed the camera control problem for single-character animations and achieved nice results, camera control for multicharacter animations has rarely been addressed so far. This problem appears to be more challenging due to the following reasons. First, the shot selection becomes more difficult as there may be concurrent events in a sequence of motion. We need a good metric to determine the importance of each event. Second, camera motion planning becomes more complicated as the number of participants varies dynamically in each event; people may leave a group to join another, or a group of people may dynamically form or dismiss from time to time. The interaction among individuals and groups often produces a complex motion, which causes camera motion planning to be very challenging since there would be frequent occlusions between characters and the environment. We need to take these issues into account to generate smooth camera motion and transition.

In this paper, we propose a camera motion planing approach that can generate an overview video of interesting events in multicharacter animations (Fig. 1). We focus on the discovery of events involving interactions among multiple characters within a close distance. To successfully discover events as well as important participants of an event, a computational model that can analyze the interaction among characters is crucial; however, it is very difficult, if not impossible, to analyze various types of the

- *I.-C. Yeh, T.-Y. Lee, and H.-J. Han are with the Department of Computer Science and Information Engineering, National Cheng Kung University, CSIE No. 1, Daxue Rd., East Dist., Tainan City 701, Taiwan, R.O.C. E-mail: ichenyeh@gmail.com, tonylee@mail.ncku.edu.tw, a13578642@gmail.com.*
- *W.-C. Lin is with the Department of Computer Science, National Chiao Tung University, 1001 University Rd., Hsinchu City 300, Taiwan, R.O.C. E-mail: wclin@cs.nctu.edu.tw.*
- *J. Lee and M. Kim are with the School of Computer Science and Engineering, Seoul National University, 1 Kwanakro, Seoul 151-744, Republic of Korea. E-mail: jehee@cse.snu.ac.kr, mmkim@mrl.snu.ac.kr.*

Manuscript received 2 Sept. 2010; revised 8 Sept. 2011; accepted 16 Oct. 2011; published online 28 Oct. 2011.
Recommended for acceptance by R. Boulic.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2010-09-0215.
Digital Object Identifier no. 10.1109/TVCG.2011.273.

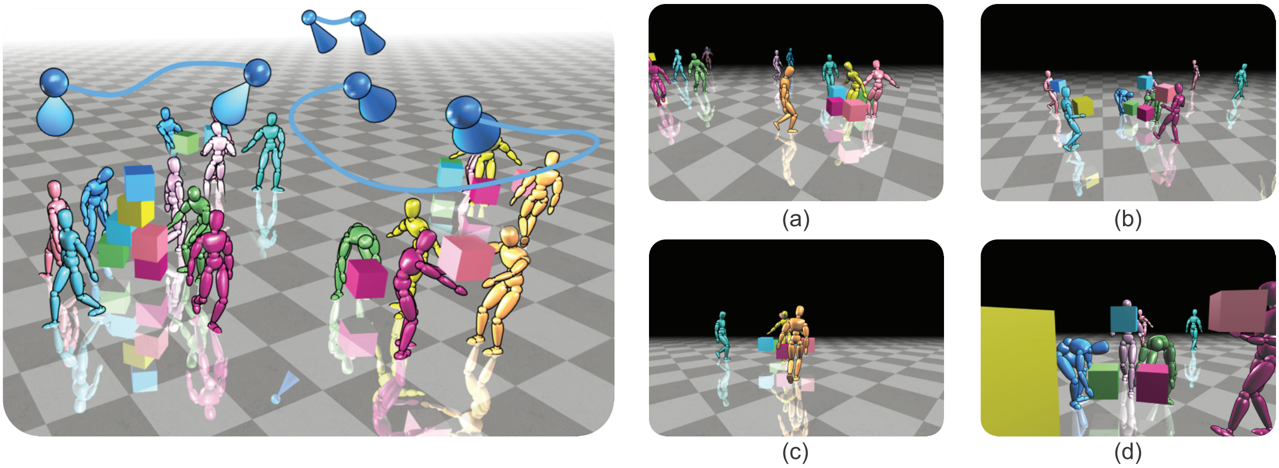


Fig. 1. Our camera control approach discovers and analyzes important events in a multicharacter animation to generate a motion overview video. (a), (b) Two snapshots of our results; (c) and (d) are the results by Assa et al. [4] at the same time frame of (a) and (b), respectively. Our results can capture multiple important events simultaneously and thus provides a clearer overview of motions.

interactions among multiple characters using a single computational model. The interactions may be local or even subtle, such as chatting with rich facial expression and gesture movement, or even an eye contact; the interaction may be global, such as a group of people gathering, marching, or separating. Interactions are often initiated with the gathering of people or involved with movements of people. This suggests that the relative position and motion between characters is very important information for describing interactions and discovering important events in multicharacter animations. In fact, several computational models based on the relative position of people have been proposed to describe the interaction between a group of people in sociological studies [12], [14], [17]. Although these models do not take local interactions into account (such local or subtle interactions are beyond the scope of this paper), it has been shown that many interactions in multiperson motions can be represented by these models. Therefore, we apply the studies of Proxemics [14], social force [17], and social network analysis [12] to analyze the dynamic relation among events and the relation among the participants within each event. For the above reason, we refer to events that can be characterized by the relative position of participants and hence be well described by these sociological models as *social events*.[1]

By analyzing the variation of relation strength among participants and spatiotemporal correlation among events, we are able to discover both important events and the leading participant of each event in a motion clip. In particular, we propose to use an event relation graph to model the relation among events and participants. The intuition of the event relation graph is that the importance of an event cannot be solely defined based on the event itself. Rather, the importance of the event is defined based on its influences to other events. More specifically, the importance of an event can be considered from three aspects: 1) the content of the event, 2) the importance of other events involved with this event, and 3) the importance of its participants. If we can identify the time and location that have large variations in these aspects, we can discover multiple important events that may happen concurrently and generate an overview video respecting these events.

The novelties of our work are proposing: 1) a sociology-based approach that utilizes techniques in social force and social network analysis to model the complex relation among individuals and events and to analyze the importance of all events in a multicharacter animation; 2) a camera control optimization approach to generate an overview video that respects concurrent social events discovered in motion clips.

Approach overview. Fig. 2 shows an overview of the proposed approach. The inputs are several motion clips in which multiple persons may interact with one another. We use the mean shift algorithm to cluster the participants of each event based on their moving trajectories. We then use the direction and location of characters in event importance analysis (Section 3). In addition, we also use motion capture data to compute the visibility information that is needed in camera motion planning. We analyze the relation among the participants of an event and the relation between events to determine the overall importance of each event at each time frame (Section 4). Based on the overall importance of events, we segment the motion clips and generate a camera control path for each split shot that optimizes for the importance of viewing events and the smoothness of camera control. The continuity between consecutive shots is handled by incorporating cinematography editing rules in the optimization process. Finally, we produce a video clip using the generated camera path (Section 5).

2 Related Work

Our work is related to the camera control problem in computer graphics, and the social force and social network analysis in sociology. As camera control and social network analysis have rich literature, it is beyond the scope of this paper to thoroughly survey both fields. Therefore, we will only review closely related work in these two fields. We refer the readers to [7] for a comprehensive survey on camera control techniques.

[1] In the rest of this paper, *event* and *social event* will be used interchangeably.

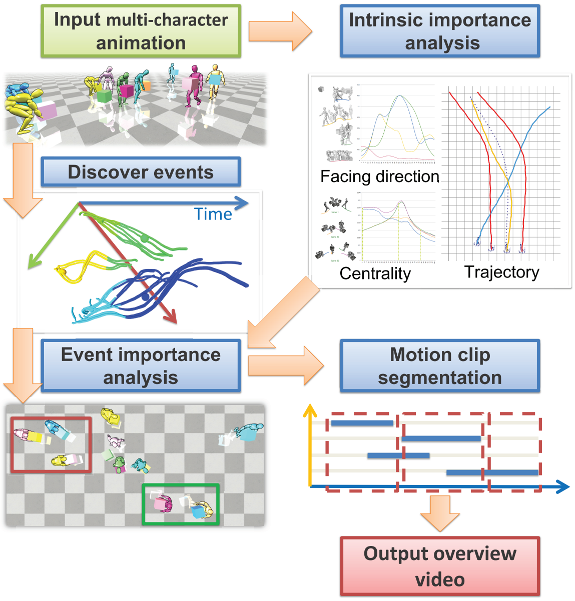


Fig. 2. Overview of our approach. We discover important events in the input animation by analyzing the dynamic relations among participants of an event and the spatiotemporal correlation among events based on related studies in sociology. An overview video is then generated based on the optimal camera path that respects event importance and maintains smooth camera control.

2.1 Camera Control

Camera control was originally studied in cinematography where a set of standard rules of camera configurations and transitions, such as *don't cross the line*, *avoid jump cuts*, *use establishing shots*, and other principles, has been developed over the years [21]. Following these rules, researchers in computer graphics, artificial intelligence, and robotics studied the camera control problem from the computing perspective. For instance, He et al. [16] constructed a state machine to describe cinematography idioms to control a camera. Several constraint-based approaches [5], [11], [15], [26] that treat cinematography principles as different camera movement constraints were also proposed. Recently, Vieira et al. [30] introduced intelligent design galleries, which apply supervised learning techniques to generate good camera positions for various scenes. Turkay et al. [29] proposed a camera placement approach for simulated crowd motion. They determine the interest points of the camera by simply analyzing the 2D positions and velocities of individuals in a crowd using information theory. In general, although most of these studies address the interaction between scene actors and the cinematic idioms, they do not pay attention to the events created by a group of people, which is a focus of our work.

Camera control approaches designed for human motions have been proposed recently [2], [4], [23]. Kwon and Lee [23] introduced a camera control technique for character animation. They focus on selecting a series of static camera positions and then interpolating them to generate a camera path. Assa et al. [4] proposed a global optimization approach to compute the required camera movement along all scene frames while satisfying the camera motion constraints. They also handle multiple shots conditions, while considering the pose saliency to better illustrate the significant poses in motion. Kardan and Casanova [20] addressed camera control for groups of multiple characters using cinematographic rules; they focused on conversational agents, in which audio input is needed to extract events. Lino et al. [25] planned a sequence of shots based on the events given by the input of narrative elements, which provide a description of the actions performed in the environment. An online view selection system for motions involving two or few characters was proposed by Assa et al. [2]. Their approach computes the correlation between the animation and the camera output and selects views that have high correlation. The correlation-based approach works effectively for motion with a few characters, but it does not scale easily to deal with a large group of characters. Though our camera control method is also based on an optimization formulation similar to Assa et al. [4], our approach is fundamentally different as we need to generate a motion overview of many characters. Specifically, the objective function in our camera control optimization problem respects concurrent salient events using techniques developed in social force and social network analysis.

Video and animation summarization has also been gaining attention in recent years. Assa et al. [3] utilize the detection of saliency for fast browsing a movie. Similar work for selecting key frames for the purpose of summarization has also been demonstrated using image features in [10]. Cheong et al. [6] converted game logs into a plan data structure and selected important action from summarized logs to produce gameplay videos. Besides, a visualization approach that depicts the variations between different MOCAP data has been proposed recently [18]. In this paper, although we adopt a similar concept of detecting salient events, we generate a continuous, time-varying overview of multicharacter animation rather than a summary consisting of image snapshots.

2.2 Social Force and Social Network Analysis

Our approach on event importance analysis mainly benefits from the studies on social force and social network analysis. The social force concept was introduced by Lewin [24] who applied the field theory into the social context and suggested that the behavior among a group of people is guided by *social forces*. In particular, our definition of relation strength between individuals is inspired by the social force model used for describing pedestrian dynamics [17], [19] and detecting abnormal crowd behaviors [27]. Social network analysis views social relationships among individuals in terms of the theory of networks consisting of nodes and ties [12]. In social network analysis, *centrality* gives a rough measurement of the social power of a node based on how well they connect the network [32].

3 Event Discovery

An event occurs when there are interactions generated among a group of people, which we call the participants of the event. A participant of an event can be a person or an object. Since an event is formed by participants, we can discover an event by grouping the participants of the event. We analyze each participant's trajectory in the input motion clips, and find the participants of each event by clustering the trajectories of all participants. To utilize higher level information of participants, we propose a two-pass clustering approach. At the first pass, trajectories of individuals and objects are sorted into groups according to their spatiotemporal distance (3D position + 1D frame index).

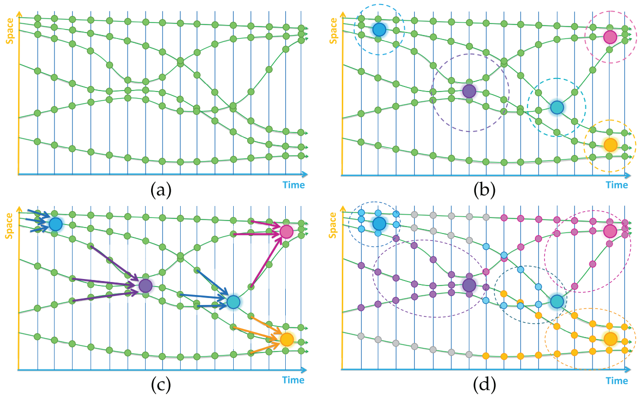


Fig. 3. Illustration of our two-pass clustering approach. (a) Trajectories of all participants plotted in a space-time space (assuming 1D time and 1D space for simplicity). (b) The first-pass clustering result using the mean-shift algorithm. (c) The destination of each participant is included in the second-pass clustering. (d) The second-pass clustering results. In (d), nodes color coded with same color belong to the same cluster representing a particular event. Note that some nodes are not clustered into any event and are color coded with gray.

At the second pass, we refine the clustering of participants by including their destination information that is acquired from the first pass. As we do not know the destination of each participant initially, we can only include the destination information after the first-pass clustering.

Fig. 3 illustrates our idea. Each green curve represents the trajectory of a participant, where each node denotes one's position at a time frame. For simplicity of illustration, we assume each participant moves in a 1D space, but in our implementation, these trajectories actually exist in a 4D space (3D position plus time). We use a feature vector to represent the property of each node and cluster all nodes in the input motion using the adaptive mean-shift algorithm [28], which is more efficient in the high-dimension clustering. At the first pass, the feature vector of a node is a 4D vector storing the 3D position and frame index. Fig. 3b shows the clustering results of the first pass, where each dashed circle is a cluster representing the spatiotemporal range of an event. The solid circle inside each cluster denotes the center of each cluster, which is considered as the destination of a participant as shown in Fig. 3c. Note that the destination vector always points forward temporally (or causally), i.e., when a participant passes an event center, its destination is the center of the event it is going to join. At the second pass, the feature vector of a node is a 7D vector including the position, destination, and frame index. The mean-shift clustering results are illustrated in Fig. 3d. In this paper, an event can be regarded as a social gathering or activity at a specific time period and spatial region. After clustering, each cluster has its temporal and spatial boundaries, and it represents a social event consisting of several participants, which are denoted by nodes with the same color in Fig. 3d, i.e., a distinct color for each cluster (event). Including the destination information in the second-pass clustering produces better clustering results. Fig. 4 shows an example of clustering results without and with destination information. Curve segments in the same color represent the trajectories of all participants of the same event. One can observe that the clustering results with destination (the cyan and yellow curves at right image) are better than those without including destination information.

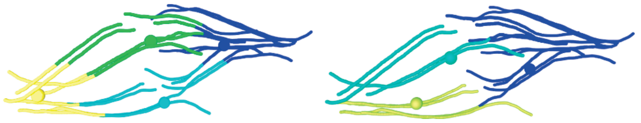


Fig. 4. Comparison of the clustering results without and with destination information. Curve segments in the same color represent the trajectories of all participants of an event. The clustering results with destination (cyan and yellow curves at right image) are better than those without including destination information.

Although our event discovery approach works well for most motion clips in our examples, there are occasional situations that the clustering results are not totally satisfactory, e.g., at the boundary frames of events. In those situations, we optionally allow a user to edit the range of each event by modifying the clustering results. This user intervention is minimal and only used in the boundary frames in few cases. For the rest of our approach, no additional user intervention is needed.

4 Automatic Event Importance Analysis

We compute the importance of an event by accumulating the importance from the local to global scope. At the local scope, the intrinsic importance of an event describes the relation changes among characters. The social force means the strength of influence between participants. The variation of social force reflects the relation changes. Therefore, the intrinsic importance is obtained by analyzing the variations of social forces among all participants of the event. At the global scope, we consider the interactions across different events by observing the spatiotemporal overlap of events. The importance of an event is reinforced if the event has large overlap (or strong correlation) with other events spatially or temporally. This is similar to a voting process in which events with interactions vote for each other to increase their importance. Given the reinforced intrinsic importance of all events, we also define the importance of a participant by summing the reinforced importance of all events joined by the participant. This helps to locate important participants within an event.

Besides the interactions among events measured by their spatiotemporal overlaps, we also consider the interaction/correlation between important participants and important events. The assumption is that if a participant joins more important events, then the participant is more important and vice versa. Hence, we compute the overall importance of the event by summing the importance of all its participants. We define the intrinsic importance E_I and the overall importance E_O of an event in the following sections.

4.1 Intrinsic Importance of an Event E_I

We evaluate the intrinsic importance of an event based on the variations of social forces in the event since the behavior of a group of people are affected by social forces among them [24]. Social forces are originally modeled as a force field called social field. As it is not easy to directly measure the social field for complex interactions among a group of people, we instead measure social forces by observing their effects on each participant of an event. Inspired by the social force model for pedestrian dynamics [17], [19], we use the facing direction of each participant to represent the potential walking direction and the mutual distance between two participants to model social forces as they are the most significant effects caused by social forces. In addition, we compute the trajectory similarity of individuals to model the group behaviors. The intuition is that the members of a group of people usually have similar trajectories within a period of time even though some members of the group may not be within a close distance. Trajectory similarity is also useful for modeling important events that exhibit long-range interactions, e.g., a character followed by other characters in a distance. This kind of long-range interaction cannot be modeled by facing direction and mutual distance, so we take trajectory similarity into account when measuring social forces.

We follow [4] to use the shoulder orientation of a human participant and the tangent vector of the trajectory of an object participant to compute the facing direction of a participant i at frame t, $\vec{f}_i(t)$. Following this definition, the variation of facing direction $\theta_i(t)$ is the angle between $\vec{f}_i(t)$ and $\vec{f}_i(t-1)$. The mean variation of facing direction caused by social forces of an event within a time window centered at frame t is then defined as

$$F(t) = \sum_{i=1}^{N} \frac{1}{2H} \sum_{m=t-H}^{t+H-1} \frac{\theta_i(m) + \theta_i(m+1)}{2} \cdot e^{-\frac{(m-t)^2}{2\sigma^2}}, \qquad (1)$$

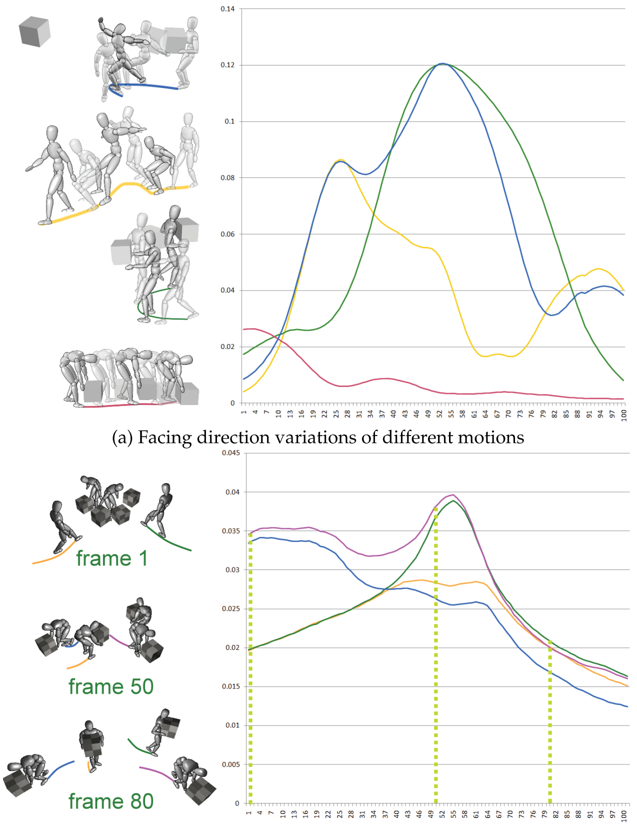
where N is the total number of participants of the event and the length of time window is $2H$. Fig. 5a shows the facing direction variations of different motions (events of a single participant). The box throwing motion has the largest variation since the participant's facing direction changes rapidly, while the box pushing motion has the smallest variation. This example shows that facing direction variation is a good measurement of social force variations.

Furthermore, we adopt the studies in social network analysis to model the relationship among the participants of an event using an undirected graph called *relation graph* [12]. The nodes and edges in the relation graph represent the participants and their relationship, respectively. Each edge stores the relation strength between two participants it connects. We define the relation strength based on the Proxemics study introduced by Hall [14]. According to Proxemics, the influence of two people on each other is inversely proportional to the square or even the cube of their mutual distance. For simplicity, we define the relation strength s_{ij} between two participants i and j as the reciprocal of their mutual distance

$$s_{ij} = \frac{1}{\|\vec{p}_i - \vec{p}_j\| + 1}, \qquad (2)$$

where $\vec{p}_i$ and $\vec{p}_j$ are the position of participant i and j, respectively, and $\|\cdot\|$ is the L2-norm. We use the adjacency matrix $\mathbf{S}$ to represent the relation strength of the relation graph of an event. Each element of the adjacency matrix is the relation strength between two participants s_{ij} defined in (2).

Having defined the relation strength on each edge of the relation graph of an event, we can evaluate the relative importance of each participant of the event. We apply the *centrality* concept used in social network analysis [12] to measure the importance of a participant, which says that the importance of a person within a social network can be measured by the centrality of a node within a graph. There are various measures of centrality, such as degree centrality, betweenness centrality, closeness centrality, and eigenvector centrality. The degree centrality is simply the degree of a node; Betweenness centrality is the total number of the shortest paths in the graph passing through a node; The closeness centrality is defined as the mean geodesic distance. We choose to use the eigenvector centrality because it can measure the importance of a node more precisely and globally according to the social network study [13].



(a) Facing direction variations of different motions

(b) Centrality variations of different participants

Fig. 5. (a) Facing direction variations of a box-throwing motion is much larger than a jumping motion or a turning motion. (b) Centrality index variation of each participant of an event. The horizontal axes in (a) and (b) represent frame number.

Given the relation graph of an event, we can compute the eigenvector centrality of all participants by solving

$$\mathbf{Sc} = \lambda \mathbf{c}, \qquad (3)$$

where $\mathbf{S}$ is the adjacency matrix of the relation graph, and λ is the largest eigenvalue of $\mathbf{S}$. $\mathbf{c} = [c_1, c_2, \ldots, c_N]^T$ is the corresponding eigenvector in which each component c_i is the centrality of the ith node. The fluctuation of the centrality of each node can be used to measure how much information comes out in this frame. Let $c_i(t)$ be the centrality of node i at frame t. The mean centrality of a participant i within a time window centered at frame t is defined as follows:

$$\overline{c}_i(t) = \frac{1}{2H+1} \sum_{m=t-H}^{t+H} c_i(m) \cdot e^{-\frac{(m-t)^2}{2\sigma^2}}. \qquad (4)$$

We call $\overline{c}_i(t)$ the centrality index of the participant i at frame t. In Section 5, centrality index is used to segment motion clip and weight the influence of each participant to camera control optimization. Fig. 5b shows an example of how the centrality index of each participant of an event varies as their mutual distance changes. At frame 1, pink and blue participants have large centrality values since they are closely located at the center of the event region. At frame 50, as green and pink participants move toward each other and orange and blue participants leave the event center, green and pink participants' centralities increase and the other participants' centralities decrease. Finally, the blue participant is the farthest one to the other participants, so its centrality is the lowest.

The centrality of an event at frame t is the summation of the centrality index of all participants

$$C(t) = \sum_{i=1}^{N} \overline{c}_i(t). \tag{5}$$

Similar to computing the centrality of a node (participant) defined in (2) to (3), we can compute the trajectory similarity of a participant simply by replacing (2) with a trajectory similarity metric between participant i and j and then applying the eigenvector analysis to obtain the trajectory similarity of each participant. We adopt the *Longest Common Subsequence* (LCSS) [31] as our metric for trajectory similarity since LCSS allows temporal stretching and spatial translation of trajectories, and more importantly, it is very robust to noise.

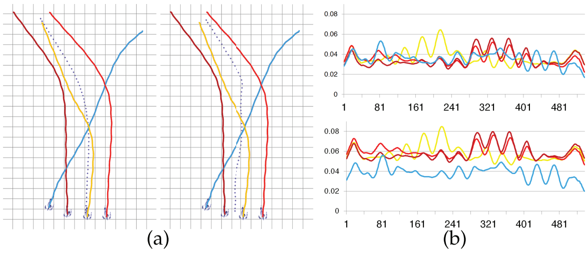
The trajectory similarity of a participant t_i reflects the similarity of a participant i's trajectory and the other participants' trajectories in an event. The trajectory similarity of an event T_s is defined as a summation of the trajectory similarity of all participants of this event

$$T_s = \sum_{i=1}^{N} t_i. \tag{6}$$

T_s represents the trajectory regularity of an event. The higher T_s is, the more similar the trajectories of the participants of this event are. Trajectory similarity is very useful for detecting a group of people performing similar motion. It can capture the overall direction of the entire group while reducing the effect of some outliers of the group motion such that a smoother target path for camera motion planning can be obtained. The target path is where we want a camera to focus on. Specifically, the shooting target of the camera at frame t is defined as a weighting combination of the position of participants

$$T_p(t) = \frac{\sum_{i=1}^{N} t_i \cdot \vec{p}_i(t)}{\sum_{i=1}^{N} t_i}, \tag{7}$$

where t_i is the trajectory similarity of participant i. A participant that does not follow the group will be excluded from the computation of camera target path. Fig. 6 shows the differences of the target path and the importance of participants with and without computing trajectory similarity. One can observe that the target path obtained with trajectory similarity (left in Fig. 6a) is smoother because the blue trajectory, which only intersects with the other trajectories at a short period, is separated and excluded from target path computation (bottom in Fig. 6b).



(a) (b)

Fig. 6. (a) Trajectory similarity of four participants. The trajectory with more reddish color has higher similarity with the trajectory of other participants. The dotted lines in the left and right plot are the target paths obtained with and without considering trajectory similarity, respectively. The target path is smoother when the trajectory similarity is considered. (b) Importance of each participant in (a). Top: without measuring trajectory similarity; Bottom: with trajectory similarity, the blue trajectory is clearly separated from the others.

An event at a time frame is considered to be more important if it has larger social force variations, larger centrality, or higher trajectory similarity. Therefore, we define the intrinsic importance of an event at a frame t as follows:

$$E_I(t) = \alpha F(t) + \eta C(t) + T_s, \tag{8}$$

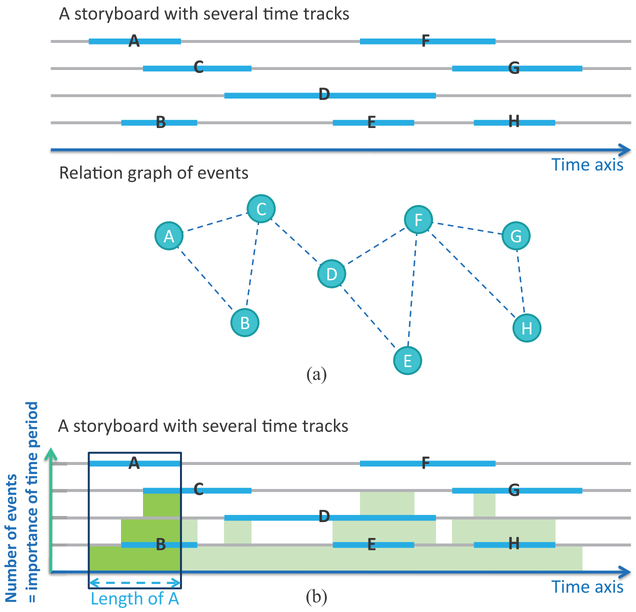
where $F(t)$ and $C(t)$ are the variation of social forces and the centrality of an event at frame t, respectively, and T_s is the trajectory similarity. The weighting of $F(t)$, $C(t)$, and T_s represents the preference of three types of group behaviors. Increasing α would increase the importance of the events with more individual behaviors, e.g., jumping, box throwing, etc. Increasing η would force the camera to focus on the group with more dynamic formation. The last one, T_s, represents the regularity of a group. Increasing the weighting of T_s causes the camera to target on those events of which participants have similar trajectories. These weights influence the definition of "important events." In our implementation, $F(t)$, $C(t)$, and T_s are normalized to $[0, 1]$ and then scaled by α and η so that their means are the same. In this way, α and η are automatically set and these three types of group behaviors can all be discovered.

4.2 Overall Importance of an Event E_O

The key idea of the overall importance measurement is voting. According to the interaction and relation between each event, we let every event vote each other to determine their relative importance. Similar to the relation graph of an event introduced in Section 4.1, we can also use a relation graph to model and measure the spatiotemporal correlation among events. To distinguish with the relation graph among participants, we will call the relation graph among events the *event relation graph*.

4.2.1 Temporal Correlation E_T

Time is an important factor of an event. To determine the temporal importance of events, we consider the overlapping of events along the time axis. Intuitively, if there are many events happening at the same time, this time period should be important since we need to pay more attention when


A storyboard with several time tracks
A
F
C
G
D
B
E
H
Time axis
Relation graph of events
A
C
B
D
E
F
G
H
A storyboard with several time tracks
Number of events = importance of time period
A
F
C
G
D
B
E
H
Length of A
Time axis

(a)

(b)

Fig. 7. (a) We view all events in a motion clip arranged on a storyboard with several time tracks and construct an event relation graph. (b) We measure the temporal correlation of events that overlap on the time axis.

there are more events happening simultaneously. For example, a fighter defeats four enemies within a short period is more important than defeating two enemies at the same time. Or at the end of performance, all people applauding at the same time is an important event.

Fig. 7 illustrates our idea of measuring the temporal correlation of events using the event relation graph. We arrange all events in a motion clip on a storyboard with several time tracks as shown in Fig. 7a, in which each alphabet denotes an event. If two events overlap temporally, we connect them with an edge weighted by the length of the overlap time. By connecting all events that have time overlap, we can construct the event relation graph of the events on the storyboard as shown at the bottom of Fig. 7a. We compute the temporal correlation of an event by taking the temporal average of the intrinsic importance of all events that have time overlap with the event. Specifically, the temporal correlation of event i is defined as follows:

$$E_T(i) = \frac{1}{T_e^i - T_b^i} \sum_{t=T_b^i}^{T_e^i} \sum_{j=1}^{J} E_I^j(t), \tag{9}$$

where T_b^i and T_e^i are the beginning and end frame number of the event i, respectively. $E_I^j(t)$ is the intrinsic importance of the jth event. J is the total number of events taking place during the time span $[T_b^i, T_e^i]$. Note that the intrinsic importance of an event is zero when it does not take place, i.e., $E_I^j(t) = 0$ if $t < T_b^j$ or $t > T_e^j$. Fig. 7b illustrates the idea of the computation of temporal correlation of the event A, where the green bars inside the time window of event A represent the intrinsic importance of all temporally overlapped events. In this case, there are three events taking place at the same time ($J = 3$). We sum up the area of these bars and divide the sum by the length of the time window to obtain the temporal correlation of the event A.

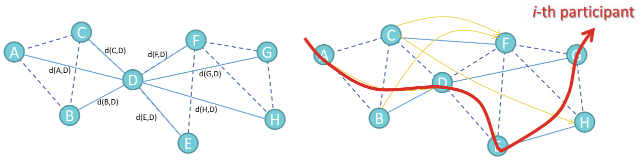


Fig. 8. (a) Spatially-correlated events vote each other to reinforce the importance of important events. (b) The importance of a participant is also affected by the importance of events that the participant has joined.

4.2.2 Spatial Correlation E_S

If there are many events occurring in the same region, this region should be important. Hence, the spatial importance of an event increases if the "event density" at which the event takes place increases. For instance, an event happens near the basket is more important than midcourt in a basketball motion clip. Similar to the computation of temporal correlation, we can compute the spatial correlation of an event i by summing the intrinsic importance of all events occurring nearby

$$E_S(i) = \sum_{j \in M(i)} E_I^j \cdot e^{-\frac{d(i,j)^2}{2\sigma^2}}, \tag{10}$$

where $M(i)$ is the set of all events linked to i event in the event relation graph. E_I^j is the summation of the intrinsic importance of event j over the duration of the event. $d(i, j)$ is the distance between the center positions of events i and j. Fig. 8 illustrates the computation of spatial correlation.

4.2.3 Overall Importance E_O

Because an important participant will participate in important events and vice versa, we compute the importance of a participant by summing the importance of all events joined by the participant. On the other hand, the importance of an event is determined based on who participates in the event. The importance of participant i is defined as follows:

$$E_P(i) = \sum_{j \in M(i)} E_T(j) + \beta E_S(j), \tag{11}$$

where $M(i)$ is the set of all events participated by the participant i, and β is a constant to weight the temporal and spatial correlation term. Finally, we measure the overall importance of an event by summing the importance of all of its participants,

$$E_O(i) = \frac{1}{N_i} \sum_{j=1}^{N_i} E_P(j) \frac{L_j}{L_i}, \tag{12}$$

where N_i is the number of participants of event i, L_j is the duration that the jth participant joins the event, and L_i is the duration of the event i.

5 Camera Control Optimization

Camera control for multicharacter animation needs to fulfill several requirements. First, the camera needs to focus on the core of the most important event while also being able to see other important events concurrently as much as possible at the background. To achieve this, we compute a shooting target of the camera at each frame using (7). This helps people understand the spatial relation between each event and obtain an overview of the events in the motion. Second, the camera shot should be illustrative. That is, we would like the camera to keep the viewing direction perpendicular to participants' moving direction. For participants not moving significantly, we prefer to see their frontal view.

We formulate our camera control problem as an optimization problem. To reduce the dimension of optimization, we divide a long motion clip into several segments and solve the optimization problem for each segment. The optimal moving path and viewing direction of the camera for each segment is obtained by minimizing the following objective function:

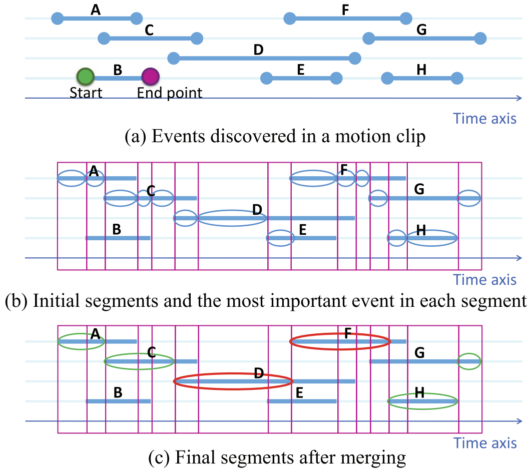
$$E = E_{internal} + \gamma E_{external} + E_{continuity}, \tag{13}$$

where $E_{external}$ is an external energy term that describes the viewpoint quality and event importance at each location and time; $E_{internal}$ is an internal energy term that enforces smoothness on the generated camera path; γ is a constant to weight the influence of internal and external energy terms. To ensure the continuity between consecutive shots, $E_{continuity}$ is added to the objective function. For the first motion segment, its $E_{continuity}$ is zero; For the other motion segments, this term is computed based on the shot continuity between the current and previous motion segments. We describe how we segment a motion clip and define the external and internal energy terms as well as the continuity term in the following paragraphs.

5.1 Motion Clip Segmentation

As our goal is to generate an overview video for the events in a multicharacter animation, it is a natural and reasonable choice to segment a motion clip according to the events occurring in the motion clip. This motivates us to generate several split camera shots for our input animation as shot splitting is a common cinematographic technique to build up an idea or change a scene. We use this technique as well as fade and dissolve to achieve a smooth transition between salient events. Furthermore, it is faster to do camera optimization for several short paths separately than a single long path. Using our event discovery approach described in Section 3, we can arrange all events on a storyboard as shown in Fig. 9a. We then use the start and end points of all events to slice the motion clip and get the initial segmentation. For each segment, we identify the most important event based on the overall importance defined in (12). The circled segments in Fig. 9b represent these most important events. Moreover, we compute the target point of the event using (7). The target point of the most important event at a segment is the target point of the segment.

After initial segmentation, we merge two short motion segments that are less than 3 seconds if they are subsequent temporally and their target points are close. The merging process repeats until there are no short segments in the overview video. For example, as the first segment of event G in Fig. 9b is less than 3 seconds and the target points of events F and G are close, we can merge the first segment of event G to the last segment of event F. The merged result is event F shown in Fig. 9c. After we merge these two segments, the optimized camera path would keep focus on event F while covering other concurrent events as much as possible until the occurrence of event H. In this way, we avoid dividing a motion clip into many small segments that cause too many split shots.



(a) Events discovered in a motion clip

(b) Initial segments and the most important event in each segment

(c) Final segments after merging

Fig. 9. (a) The events discovered in the motion clip. (b) Initial segments obtained by dividing the motion clip according to the start and end points of all events. At each segment, the most important event is circled. (c) Final segments by merging small segments that are temporally or spatially close.

5.2 Internal Energy Term

We adopt the internal force energy proposed in [4] to measure the smoothness of a camera path. Here, we briefly describe their definition of the internal energy. For better control over the camera speed, the internal energy prefers a static camera if possible. In cases where movement is required, camera speed should be as constant as possible or at least with minimal acceleration, and should be smaller than a maximal speed limit

$$\begin{aligned} E_{internal} = \sum_t & c_1(q_{t-1} - 2q_t + q_{t+1})^2 \\ & + c_2 \left\lfloor \frac{q_{t+1} - q_{t-1}}{2\dot{S}_{\max}} \right\rfloor_{>1}^4 + c_3(\alpha_{t-1} - 2\alpha_t + \alpha_{t+1})^2 \\ & + c_4 \left\lfloor \frac{\alpha_{t+1} - \alpha_{t-1}}{2\dot{\alpha}_{\max}} \right\rfloor_{>1}^4 - c_5 \left\lfloor \frac{q_{t+1} - q_{t-1}}{2\dot{F}_{\min}} \right\rfloor_{<1}^2, \end{aligned} \tag{14}$$

where the condition operator $\lfloor \rfloor$ is defined as follows:

$$\lfloor x \rfloor_{condition} = \left\{ \begin{array}{ll} x & \text{condition is true} \\ 0 & \text{otherwise} \end{array} \right\}. \tag{15}$$

q_t and α_t are the camera location and viewing angle to the target in frame t. $\dot{S}_{\max}$, $\dot{\alpha}_{\max}$, and $\dot{F}_{\min}$ are the maximal speed, angular speed, and minimal friction of the camera. In all of our examples, $\dot{S}_{\max}$, $\dot{\alpha}_{\max}$, and $\dot{F}_{\min}$ are set to $character\_height/6$, 20 degrees, and $character\_height/20$, respectively. $c_{1..5}$ are the coefficients of the different terms. In all of our examples, they are set to 10, 10, 5, 5, 2, correspondingly.

5.3 External Energy Term

The external energy term measures the viewpoint quality at each location and time. Specifically, the external energy term is the weighted sum of the viewpoint quality for all participants of all events in the entire motion clip

$$E_{external} = -\sum_{t} S_F(t) \sum_{e \in M(t)} \sum_{i \in M(e)} \bar{c}_i(t) V_i(t, q_t, \alpha_t), \tag{16}$$

where $S_F(t)$ is the frame saliency that weights the influence of frame t. It is defined as the sum of the importance of all events occurring at frame t

$$S_F(t) = \sum_{e \in M(t)} E_O(e), \tag{17}$$

where E_O is defined in (12). $\bar{c}_i(t)$ is the centrality index (defined in (5)) of participant i that weights the influence of the participant. $M(t)$ and $M(e)$ denote the set of all events at frame t and the set of all participants of event e, respectively. $V_i(t, q_t, \alpha_t)$ is the viewpoint quality of the camera at position q_t and viewing angle α_t for participant i at frame t.

The viewpoint quality of a participant is decided based on several attributes, including visibility, frontal view, perpendicular moving, and best viewing distance. The visibility attribute E_{vis} is determined by the area inside the convex hull of a participant projected on the image plane. It encourages camera to shoot where significant characters are most visible. The frontal view attribute E_{front} prefers the camera viewing the frontal face of the participant. The perpendicular moving attribute E_{move} favors the camera viewing direction that is perpendicular to the moving direction of the participant so that the most significant movement would be best seen. Finally, the viewing distance attribute E_{dist} keeps the distance between the camera and the participant from being too close or too far. This attribute only takes effect when the distance to the participant $dist(i)$ is out of a desired range $[dist_{near}, dist_{far}]$. The following equations summarize these attributes adopted in our viewpoint quality for the participant i at frame t:

$$V_i(t, q_t, \alpha_t) = c_6 E_{vis} + c_7 E_{front} + c_8 E_{move} + c_9 E_{dist}. \tag{18}$$

$$E_{vis}(i, q_t, \alpha_t) = ConvexHullArea(i) \tag{19}$$

$$E_{front}(i, q_t, \alpha_t) = -\vec{v}_{cam} \cdot \vec{f}_i \tag{20}$$

$$E_{move}(i, q_t, \alpha_t) = 1 - \|\vec{v}_{cam} \cdot \vec{v}_i\| \tag{21}$$

$$\begin{aligned} E_{dist}(i, q_t, \alpha_t) = & \lfloor dist_{near} - dist(i) \rfloor_{dist(i) < dist_{near}} \\ & + \lfloor dist(i) - dist_{far} \rfloor_{dist(i) > dist_{far}}, \end{aligned} \tag{22}$$

where $c_{6..9}$ are the weighting coefficients of the various terms. $\vec{v}_{cam}$ is the viewing direction of the camera; $\vec{f}_i$ and $\vec{v}_i$ are the facing direction and moving direction of the participant i. Note that all these attributes are time varying and depend on the camera position q_t and viewing angle α_t, but we omit variables t, q_t, and α_t in all of these equations for simplicity of notations.

We use the simulated annealing method to compute the camera path by minimizing the energy function in (13). The initial guess is obtained as follows: For each motion segment, we create a bounding box that can cover the region of all events in the starting frame of the segment. We then uniformly distribute sample points inside the box as static cameras and find the best location as the initial guess for the camera control optimization.

5.4 Continuity between Shots

To ensure the shot continuity between motion segments, we incorporate editing rules of cinematography in the optimization process. First, we avoid jump cut between consecutive shots by adding a penalty term in the objective function. The basic idea of jump cut avoidance is to make spatial discontinuity of characters unnoticeable. The rule prevents an object from appearing in two consecutive shots with a similar but different view (shooting angle) as human eyes can easily track the object's motion and notice any discontinuity caused by small view variations. On the contrary, if the shooting angles of two consecutive shots differ a lot, human eyes would not be able to track the object's motion easily and be less likely to notice the discontinuity. Therefore, we restrict the variation of viewing angles between consecutive shots to be larger than 30 degrees to prevent the jumping effect between shots. This 30-degree rule is suggested by Corrigan and White [8] as "the transition between two shots less than 30 degrees apart might be perceived as unnecessary or discontinuous." Second, we enforce the 180-degree rule, which says that the camera movement should keep at the same side of main characters and preserve the consistent viewing direction of group motion [1]. For example, two characters in the same scene should always have the same left/right relationship to each other in two consecutive shots. Similar to the jump cut penalty, we penalize the displacement of camera in consecutive shots that cross the 180-degree line. Formally, we define $E_{continuity}$ as follows:

$$E_{continuity} = E_{jump} + E_{180}, \tag{23}$$

where

$$\begin{aligned} E_{jump} &= \lfloor \infty \rfloor_{\frac{\vec{v}_{cam}(t-1) \cdot \vec{v}_d}{\|\vec{v}_{cam}(t-1)\| \|\vec{v}_d\|} > \cos(30^\circ)}, \\ E_{180} &= \lfloor \infty \rfloor_{(\vec{v}_{cam}(t-1) \times \vec{v}_e(t-1)) \cdot (\vec{v}_{cam}(t) \times \vec{v}_e(t-1)) < 0.} \end{aligned}$$

$T_p(t)$ is defined in (7). As shown in Fig. 10, $\vec{v}_{cam}(t) = q_t - T_p(t)$ is the viewing direction of frame t. $\vec{v}_d = q_t - T_p(t-1)$ is a directional vector from target position of frame $t-1$ to the camera position of the first frame of the second shot. We constrain that the angle between the first and second shots is larger than 30 degrees and the camera positions, q_{t-1} and q_t, must be on the same side of the motion direction $\vec{v}_e(t)$ defined by

$$\vec{v}_e(t) = \frac{\sum_{i=1}^{N} t_i \cdot \vec{f}_i(t)}{\sum_{i=1}^{N} t_i}, \tag{24}$$

where t_i and $\vec{f}_i(t)$ are defined in (6) and (20). $\vec{v}_e(t)$ is the weighted average of the facing directions of all participants of a targeted event. It represents the motion direction of the event at frame t.

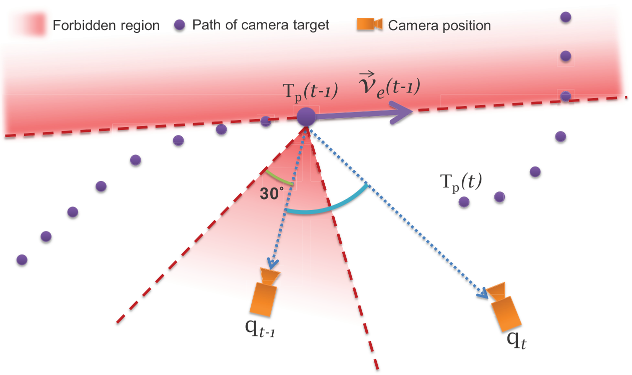
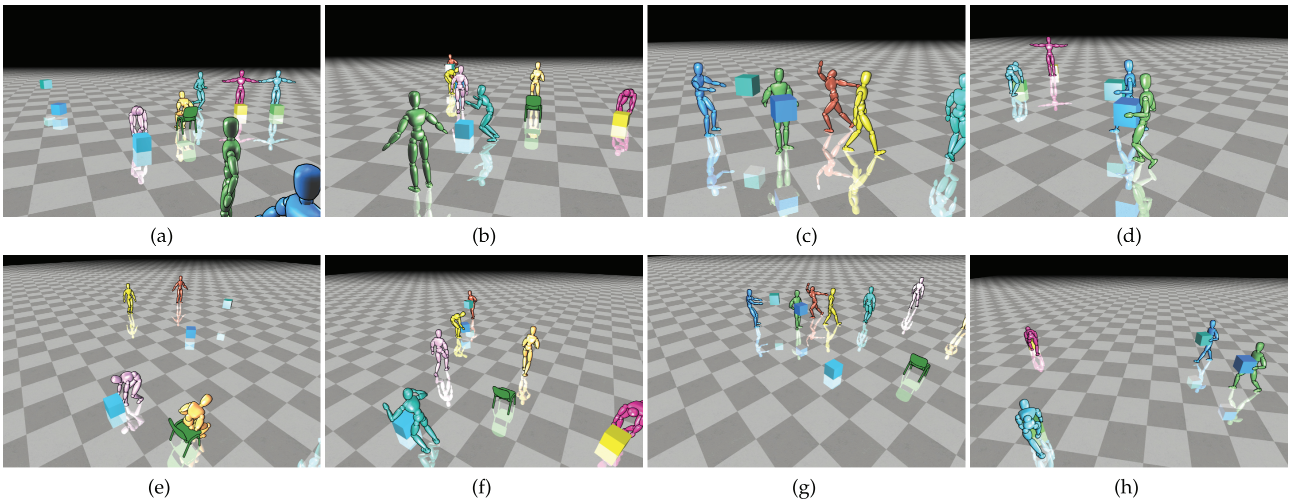


Fig. 10. Illustration of the constraints between two consecutive shots. The violet dotted lines are the target path of the camera. $\vec{v}_e(t-1)$ is the weighted average of facing directions of all participants of the targeted event. The red space is a forbidden region for the camera path of the second shot starting at frame t.

6 Results and Discussion

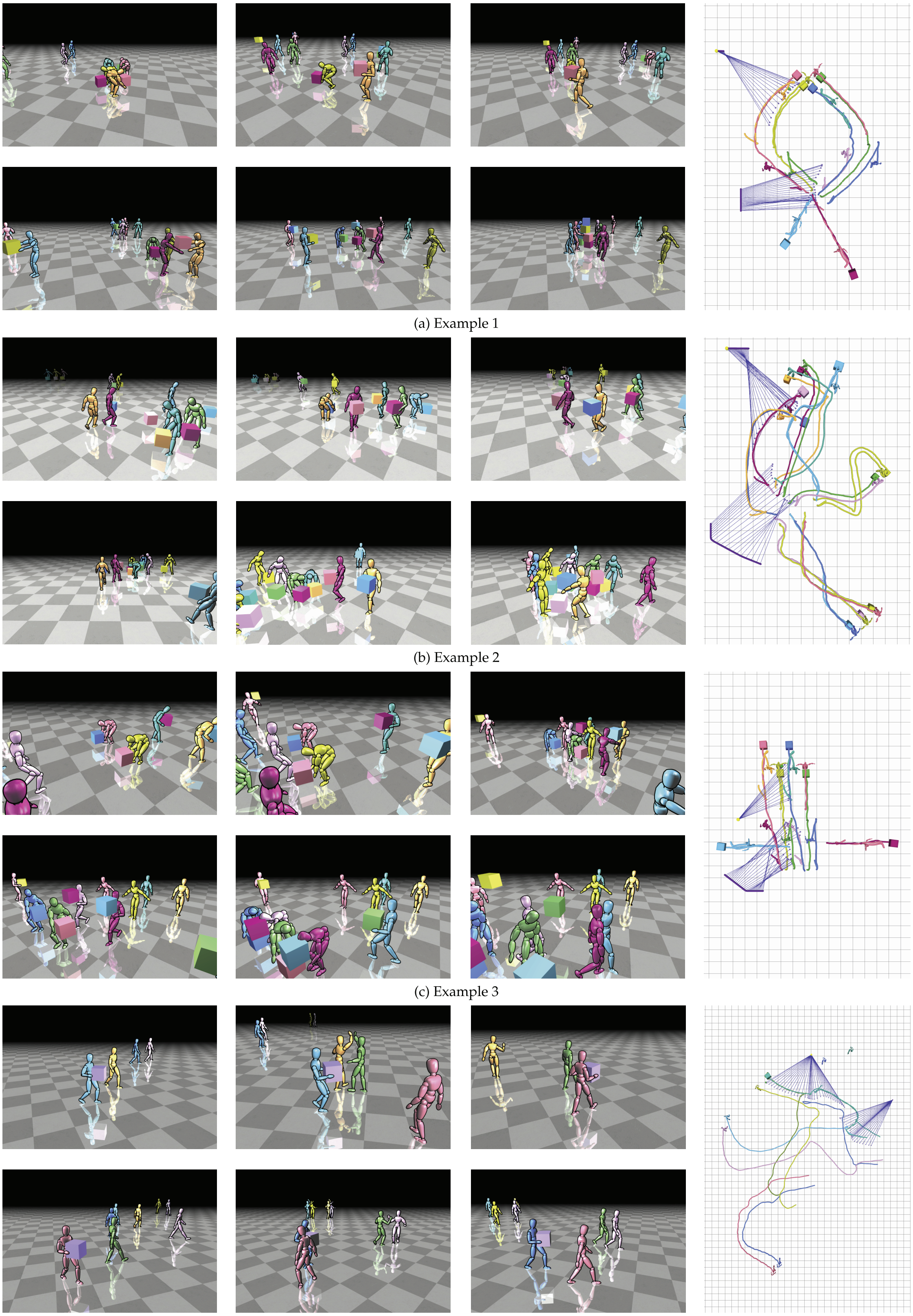
We tested our approach on multicharacter animations with different types of interactions, ranging from simple tasks such as one character passing an object to another, to complex group behaviors involving many characters interacting with one another and the environment. As it is more expensive to capture multicharacter motions, to increase the variety of data, we also use the multicharacter motion editing approach [22] to generate some motion clips from MOCAP examples. To validate the effectiveness of event importance analysis, we generate two types of motion overview videos obtained with and without event importance analysis. Videos without event importance analysis are produced by setting the weighting factor of each participant as its trajectory similarity, i.e., treat all characters in an animation as the participants of the same event. Figs. 11a, 11b, 11c, and 11d show four snapshots taken from the video generated with event importance analysis while (e)-(h) show the snapshots of the video generated without event importance analysis at corresponding time frames. In Fig. 11a, our approach shoots toward the facing direction of the pink character and focuses on a box pushing event while the video without event importance analysis in Fig. 11e does not capture this event at a good viewing angle and distance. In Fig. 11b, our approach not only focuses on the active event (a jumping blue character) but also covers the entire activity properly; however, in Fig. 11f, the shot is pointless. In Fig. 11c, our approach generates a good shot to illustrate two important events (box passing and box throwing between characters). On the contrary, without event analysis, the box throwing motion cannot be captured well since it is far from the camera as shown in Fig. 11g. Similarly, Figs. 11d and 11h demonstrate that our approach can target at important events and provide a better shot than the approach without event importance analysis. For better demonstration of our results, please see a comparison in the main video, available in the online supplemental material.

Fig. 12 shows four of our results on motion clips of multiple characters and objects. In Example 1, the camera first focuses on two events: box passing and box pushing between two characters. Once the box is passed, the camera switches its focus to characters stacking boxes in the scene. This example illustrates the interaction between important participants and events as shown in Fig. 8b. The importance of the box is reinforced as it “participates” in several important events (being carried by different people); In Example 2, there are 10 characters carrying boxes in the motion clip. These characters form several groups. One can see that our camera control approach tracks the largest group and selects a good view to cover the entire crowd successfully; In Example 3, several characters push, move, and stack boxes. Our approach presents the major event clearly in two shots. Example 4 shows nine characters forming several social events dynamically. Our approach identified several important events, such as high-five and box passing between characters. Moreover, all camera shots follow the editing rules of 180-degree and avoiding jump cut. From our results in Figs. 11 and 12, it appears that our approach can successfully trace important events in motion


(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 11. (a), (b), (c), and (d) are three images, respectively, taken at the 6th, 11th, 17th, and 23rd second of the motion overview video generated by our approach. (e), (f), (g), and (h) are images taken at corresponding time of the video generated without event importance analysis. Please see the main video, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2011.273, for better demonstration.



(a) Example 1

(b) Example 2

(c) Example 3

(d) Example 4

Fig. 12. Snapshots of some of our results. The rightmost image shows the trajectories of participants and camera paths.

clips and generate smooth camera paths as well as reasonable shot transitions. An important feature of our camera control approach is that it can simultaneously capture multiple events since the objective function in (16) sums the importance of all events at a time frame. This feature is best illustrated in Examples 1 and 4. Please also see the accompanying videos, available in the online supplemental material, for these results.

TABLE 1
Pairwise Voting Results in Each Aspect

	Stable	Professional	Narrative	Total (2520)
1. Our approach	340	315	336	991
2. Artist	345	335	273	953
3. without event analysis	155	190	231	576

User study. We also conducted a user study to evaluate the smoothness and comprehensibility of camera control in our results. In the first part of the study, we compare our results with overview videos generated without event analysis and by an artist. We tested on four animation data, so there is a total of 12 motion overview videos. We performed our user study using the method of paired comparisons [9] for the statistical evaluation of subjective preferences. In this method, items are presented side-by-side in pairs to a human subject, who then selects a preferred one in each pair. Following this method, we prepared a web-based survey showing pairs of motion overview videos. For each pair, a subject needs to vote which one is more stable, professional and narrative. There were 70 subjects participating in our user study including 60 males and 10 females, 9 (of 70) subjects have art background. Each subject watched 12 pairs of videos, so each approach was compared $4 \times 3 \times 70 = 840$ times. To reduce the bias in our user study, we randomized the playing order of video pairs and provided only the most necessary information to the subjects.

Table 1 lists the pairwise voting results in each aspect. One can find that our approach is the most narrative while being comparably as stable and professional as artist-made videos. Table 2 shows the overall preference by summing the votes in all aspects for each method across subjects. For example, in a comparison of methods 1 and 2, if method 1 is voted by a subject in the stable and professional aspects while method 2 is preferred in the narrative aspect, then the preference times of method 1 and 2 are 2 and 1, respectively. The preference times from different subjects for each method are summed to obtain Table 2. It shows that our approach is clearly preferred over the camera control without event analysis since our results were favored in 66.19 percent (556 of 840) of the comparisons with the camera control without event analysis. When compared with the professional results by an artist, our results were still preferred in 51.79 percent (435 of 840), while the results without event analysis were only favored in 34.76 percent (292 of 840). Overall, the subjects favored our method in 39.33 percent (991 of 2,520) of the comparisons, while the preference for the artist's results and the results without event analysis are 37.82 percent (953 of 2,520) and 22.86 percent (576 of 2,520), respectively. The intraobserver variability, Kendall's coefficient of consistence $\zeta \in [0, 1]$, had a very high average of $\zeta = 0.88$ and a small standard deviation $\sigma = 0.15$. This indicates that each subject is rather consistent when making his/her choices. The interobserver variability, Kendall's coefficient of agreement, is $u = 0.065$ for Table 2, with a p-value <0.01. Hence, there is a statistically significant agreement among the subjects regarding the three approaches. We refer readers to [9] for a detailed explanation of these indicators.

TABLE 2
Preferences of 70 Subjects for Three Camera Control Approaches, e.g., an Entry n in Row 1 and Column 2 Means the Result of Approach 1 Was Preferred n-Times to the Result of Approach 2

	1	2	3	Total (2520)
1. Our approach	-	435	556	991
2. Artist	405	-	548	953
3. without event analysis	284	292	-	576

TABLE 3
Correct Rate and Average Response Time in the Questionnaire Study of Comprehensibility

Motion	Method	Correct rate (%)	Avg. response time (sec)
Ex0	Ours	83.57	13.51
	Artist	81.43	14.35
Ex1	Ours	90	9.95
	Artist	85	12.15
Ex2	Ours	85.71	12.40
	Artist	80.71	11.00

In the second part of our user study, we judge the comprehensibility of our results by conducting a questionnaire survey, which asks subjects some questions related to the comprehension of a scene via our motion overview videos and artist-made videos. We asked each subject to watch a video before showing questions. We then recorded each subject's answer and response time, which is counted starting from when the questions were shown. To reduce the bias in our user study, we randomized the playing order and question sets to make sure that the same question would not be read twice by each subject. 70 subjects participated in this study. Each of them answered two questions for each video. For the details of our questionnaire, please see the online supplementary material. Table 3 summarizes the survey results. This survey attested that the results generated by our approach are quite understandable to the user. The comprehensibility of our results are also comparable to that of the artist's results.

Interactive event clustering and selection. Although our approach can generate an overview video of social events based on sociological studies, the suggested events in the overview video may not totally satisfy a user's personal preference. To enhance the usability of our system, we also develop a convenient interface to allow a user to manually edit event clustering of participants. In the accompanying video, available in the online supplemental material, we demonstrate an interface in which users can modify the event clustering results or add a new cluster to create a new event. In addition, if the user wants to see the events relevant to a specific character, an object or a specific group of participants, the user can increase the weighting of selected participants. In this way, the importance of the events related to these participants will be enhanced such that these events will be focused in the overview video. Our interface provides a great flexibility that allows a user to perform not only high-level control on event selection but also low-level editing on event clustering.

TABLE 4
This Table Lists the Computational Time of Our Approach in Different Examples with Different Types of Motions and Interactions

Motion	Content	P	F	EA (sec)	CP (sec)
Ex0	jumping, box moving / throwing	14	990	3.46	326.56
Ex1	box moving / pushing / passing	16	492	1.34	104.548
Ex2	box moving, pushing, passing	20	493	1.96	167.99
Ex3	box moving / pushing / passing	16	441	1.25	75.132
Ex4	walking, highfive	8	634	1.61	154.21
Ex5	walking, highfive	8	634	1.61	257.99
Ex6	box moving / passing	4	582	0.854	155.56
C1(1)	walking, talking	19	776	12.8	135.94
C1(1,2)	walking, talking	43	776	38.8	363.32
C1(1,2,3)	walking, talking	59	776	55.45	633.50

The content of motion, number of participants (P), and number of frames (F), event analysis time (EA), and camera planning time (CP) are listed.

Computational performance. We implemented our approach using C# code on an Intel Core i7 machine. The computational time depends on the total number of characters and frames in a motion clip. Table 4 lists the computational time of our approach in different examples with different types of motions and interactions. To test the scalability of our approach, we also tested our approach on an artificially-made crowd motion C1. There are three groups of people in example C1, where C1(1), C1(1,2), and C1(1,2,3) represent the cases that only the first group, the first and second groups, and all groups are considered in event analysis and camera control, respectively. For clips with 441-990 frames and 4-59 participants (3-59 characters, 1-10 boxes), the computational time ranges from 2 to 12 minutes. Our event analysis needs to perform all-pair computation among participants of each event several times. Thus, the computational time of event analysis depends on the number of participants and the length of each event. Besides, as evaluating viewpoint quality in the optimization process is the most time consuming, more computational time is required if the number of participants is larger or the duration of an event is longer.

Limitations. Currently, there are two main limitations of our approach. First, our approach can only generate a motion overview in a “linear” manner with respect to time. There may be many important events at the same time; however, the camera can only focus on few of them at each time instant. Therefore, in crowd animation (with hundreds or thousands of people), there will be too many events occurring if the crowd does not have high motion coherence, e.g., marching. It will be very difficult to locate events that the viewer may be interested in. Our method cannot deal with such cases. The nonlinear camera editing capability would allow concurrent events to be viewed sequentially and a single event to be viewed multiple times at different viewpoints. It is debatable whether all people would favor this style of video results though. More future work will be done toward this direction. Second, our event analysis currently cannot measure the decaying of the importance within an event. If there is no preemptive incoming event, an important long event may become overwhelming and force camera to focus on the entire event even though its importance may be decaying. We will address these issues in our future work.

7 Conclusion and Future Work

In this paper, we introduced a sociology-based approach to model the dynamic relations among individuals and events in multicharacter animation. This approach measures the importance of events and their participants. Based on the measured importance in animation, our approach automatically computes the optimal camera path that not only maintains smooth camera control but also respects event importance. We evaluate the validness and effectiveness of our approach by comparing our results with those generated without event analysis and handcrafted by an artist. The comparison shows that our approach can better capture multiple important events in animation. In addition, the user study participated by 70 subjects indicates that our results are preferred in 66.19 percent of the pairwise comparisons with those generated without event analysis and are comparable (51.79 percent) to the artist’s results. Currently, our approach does not consider individuals’ local and subtle actions to discover important events or interesting individual actions from multicharacter animation. We will consider these in our future study. We would also like to improve the computational performance and extend our approach for virtual cinematography applications that allow nonlinear-time motion overview of multicharacter animation.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments, Mr. Cheng Chou for voice recording in our supplemental videos and Mr. Yueh-Han Lin at TOKO University, Taiwan for making the artist results. This work was supported in part by the National Science Council (contracts NSC-100-2221-E-006-188-MY3, NSC-100-2628-E-006-031-MY3, NSC-99-2221-E-006-066-MY3 and NSC-100-2628-E-009-002-), Taiwan and the UST-UCSD International Center of Excellence in Advanced Bioengineering sponsored by the Taiwan National Science Council I-RiCE Program under Grant Number: NSC-99-2911-I-009-101. Besides, the fifth and sixth authors are supported in part by the National Research Foundation of Korea (No.2011-0018340). The corresponding author of this project is Prof. Tong-Yee Lee.

References

[1] D. Arijon, *Grammar of the Film Language.* Silman-James Press, 1976.
[2] J. Assa, L. Wolf, and D. Cohen-Or, “The Virtual Director: A Correlation-Based Online Viewing of Human Motion,” *Proc. Eurographics Conf.*, pp. 595-604, 2010.
[3] J. Assa, Y. Caspi, and D. Cohen-Or, “Action Synopsis: Pose Selection and Illustration,” *Proc. SIGGRAPH,* vol. 24, pp. 667-676, Aug. 2005.

[4] J. Assa, D. Cohen-Or, I.-C. Yeh, and T.-Y. Lee, "Motion Overview of Human Actions," *Proc. SIGGRAPH,* vol. 27, no. 5, pp. 1-10, Dec. 2008.

[5] W.H. Bares, S. Thainimit, and S. McDermott, "A Model for Constraint-Based Camera Planning," *Proc. AAAI Spring Symp. Smart Graphics,* vol. 4, pp. 84-91, 2000.

[6] Y.G. Cheong, A. Jhala, B.C. Bae, and R.M. Young, "Automatically Generating Summary Visualizations from Game Logs," *Proc. Fourth Artificial Intelligence and Interactive Digital Entertainment Conf. (AIIDE),* C. Darken, M. Mateas, C. Darken, and M. Mateas, eds., The AAAI Press, http://dblp.uni-trier.de/rec/bibtex/conf/aiide/CheongJBY08, 2008.

[7] M. Christie and P. Olivier, "Camera Control in Computer Graphics: Models, Techniques and Applications," *Proc. SIGGRAPH ASIA,* pp. 1-197, 2009.

[8] T. Corrigan and P. White, *The Film Experience,* p. 140. Bedford/St. Martin's, 2004.

[9] H.A. David, *The Method of Paired Comparisons.* Charles Griffin & Company, 1988.

[10] D. DeMenthon, V. Kobla, and D. Doermann, "Video Summarization by Curve Simplification," *Proc. MULTIMEDIA '98: Sixth ACM Int'l Conf. Multimedia,* pp. 211-218, 1998.

[11] S.M. Drucker and D. Zeltzer, "Intelligent Camera Control in a Virtual Environment," *Proc. GI '94: Graphics Interface Conf.,* pp. 190-199, 1994.

[12] L. Freeman, *The Development of Social Network Analysis.* Empirical Press, 2006.

[13] R. Grassi, S. Stefani, and A. Torriero, "Some New Results on the Eigenvector Centrality," *The J. Math. Sociology,* vol. 31, no. 3, pp. 237-248, 2007.

[14] E.T. Hall, *The Hidden Dimension.* Anchor Book, 1966.

[15] N. Halper, R. Helbing, and T. Strothotte, "A Camera Engine for Computer Games: Managing the Trade-Off between Constraint Satisfaction and Frame Coherence," *Proc. Eurographics Conf.,* vol. 20, no. 3, pp. 174-183, 2001.

[16] L.-W. He, M.F. Cohen, and D.H. Salesin, "The Virtual Cinematographer: A Paradigm for Automatic Real-Time Camera Control and Directing," *Proc. SIGGRAPH,* pp. 217-224, 1996.

[17] D. Helbing and P. Molnar, "Social Force Model for Pedestrian Dynamics," *Physical Rev. E,* vol. 51, no. 5, pp. 4282-4286, May 1995.

[18] Y. Hu, S. Wu, S. Xia, J. Fu, and W. Chen, "Motion Track: Visualizing Variations of Human Motion Data," *Proc. PacificVis '10: IEEE Pacific Visualization Symp.,* pp. 153-160, 2010.

[19] A. Johansson, D. Helbing, and P. Shukla, "Specification of the Social Force Pedestrian Model by Evolutionary Adjustment to Video Tracking Data," *Advances in Complex Systems,* vol. 10, no. 2, pp. 271-288, 2007.

[20] K. Kardan and H. Casanova, "Virtual Cinematography of Group Scenes Using Hierarchical Lines of Actions," *Proc. Sandbox '08: ACM SIGGRAPH Symp. Video Games,* pp. 171-178, 2008.

[21] S.D. Katz, *Film Directing Shot by Shot: Visualizing from Concept to Screen.* Michael Wiese Productions, 1991.

[22] M. Kim, K. Hyun, J. Kim, and J. Lee, "Synchronized Multi-Character Motion Editing," *ACM Trans. Graphics,* vol. 28, no. 3, pp. 1-9, 2009.

[23] J.-Y. Kwon and I.-K. Lee, "Determination of Camera Parameters for Character Motions Using Motion Area," *The Visual Computer,* vol. 24, pp. 475-483, 2008.

[24] K. Lewin, *Field Theory in Social Science.* Harper and Brothers, 1951.

[25] C. Lino, M. Christie, F. Lamarche, G. Schofield, and P. Olivier, "A Real-Time Cinematography System for Virtual 3D Environments," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation,* 2010.

[26] H. McCabe and J. Kneafsey, "A Virtual Cinematography System for First Person Shooter Games," *Proc. Int'l Digital Games Conf.,* pp. 25-35, 2006.

[27] R. Mehran, A. Oyama, and M. Shah, "Abnormal Crowd Behavior Detection Using Social Force Model," *Proc. CVPR '09: IEEE Conf. Computer Vision and Pattern Recognition,* pp. 935-942, 2009.

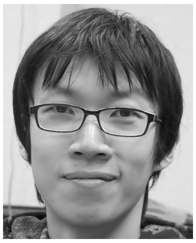
[28] *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice,* G. Shakhnarovich, T. Darrell, and P. Indyk, eds. MIT Press, Mar. 2006.

[29] C. Turkay, E. Koc, and S. Balcisoy, "An Information Theoretic Approach to Camera Control for Crowded Scenes," *The Visual Computer,* vol. 25, nos. 5-7, pp. 451-459, 2009.

[30] T. Vieira, A. Bordignon, A. Peixoto, G. Tavares, H. Lopes, L. Velho, and T. Lewiner, "Learning Good Views through Intelligent Galleries," *Proc. Eurographics Conf.,* pp. 717-726, 2009.

[31] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering Similar Multidimensional Trajectories," *Proc. Int'l Conf. Data Eng.,* pp. 673-684, 2002.

[32] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications.* Cambridge Univ. Press, 1994.



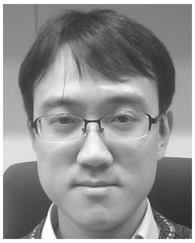
I-Cheng Yeh received the BS degree from the Department of Computer and Information Science, National Chiao Tung University, Taiwan, in 2004. He is currently working toward the PhD degree at the Department of Computer Science and Information Engineering, National Cheng-Kung University. His research interests are in the computer graphics and include mesh parameterization, shape manipulation, and camera planning.

Wen-Chieh Lin received the BS and MS degrees in control engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 1994 and 1996, respectively, and the PhD degree in robotics from Carnegie Mellon University, Pittsburgh, in 2005. Since 2006, he has been with the Department of Computer Science and the Institute of Multimedia Engineering, National Chiao-Tung University, as an assistant professor. His current research interests include computer graphics, computer animation, and computer vision. He is a member of the IEEE and the ACM.

Tong-Yee Lee received the PhD degree in computer engineering from Washington State University, Pullman, in May 1995. He is currently a distinguished professor in the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, Taiwan, ROC. He leads the Computer Graphics Group, Visual System Laboratory, National Cheng-Kung University (http://graphics.csie.ncku.edu.tw/). His current research interests include computer graphics, nonphotorealistic rendering, medical visualization, virtual reality, and media resizing. He also serves on the editorial boards of the *IEEE Transactions on Information Technology in Biomedicine*, the Visual Computer and the Computers and Graphics Journal. He served as a member of the international program committees of several conferences including the IEEE Visualization, the Pacific Graphics, the IEEE Pacific Visualization Symposium, the IEEE Virtual Reality, the IEEE-EMBS International Conference on Information Technology and Applications in Biomedicine, and the International Conference on Artificial Reality and Telexistence. He is a senior member of the IEEE and the member of the ACM.

Hsin-Ju Han received the BS and MS degree from the Department of Computer Science and Information Engineering, National Cheng-Kung University, Taiwan, in 2008 and 2010, respectively. Now, she is working in the Taiwan Semiconductor Manufacturing Company. Her research interests include computer graphics and animation.

Jehee Lee received the BS, MS, and PhD degrees in computer science from Korea Advanced Institute of Science and Technology in 1993, 1995, and 2000, respectively. He is an associate professor of computer science at Seoul National University. He is leading the SNU Movement Research Laboratory. His research interests are in the areas of computer graphics and animation. More specifically, he is interested in developing new ways of understanding, representing, and animating human movements. This involves full-body motion analysis and synthesis, biped control and simulation, motion capture, motion planning, data-driven and physically based techniques, interactive avatar control, crowd simulation, and facial animation.



Manmyung Kim received the BS degree in mechanical and aerospace engineering and the MS degree in computer science and engineering from Seoul National University. He is currently working toward the PhD degree in computer science and engineering at Seoul National University. His research interests include synthesizing character animation and artificial intelligence.