

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
7 May 2009 (07.05.2009)

PCT

(10) International Publication Number
WO 2009/055929 A1

- (51) International Patent Classification:
H04N 5/265 (2006.01) G11B 27/031 (2006.01)
G03B 35/08 (2006.01) H04N 13/02 (2006.01)
G06T 15/70 (2006.01)
- (21) International Application Number:
PCT/CA2008/001925
- (22) International Filing Date: 31 October 2008 (31.10.2008)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
61/001,109 31 October 2007 (31.10.2007) US
- (71) Applicant (for all designated States except US): XTRA-NORMAL TECHNOLOGIE INC. [CA/CA]; 5555, avenue Casgrain, 2e étage, Montréal, Québec H2T 1Y1 (CA).
- (72) Inventor; and
- (75) Inventor/Applicant (for US only): RONFARD, Rémi [CA/CA]; 899, avenue Dollard, Montréal, Québec H2V 3G8 (CA).

- (74) Agent: OGILVY RENAULT LLP; Suite 1600, 1981 McGill College Avenue, Montreal, Québec H3A 2Y3 (CA).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL,

[Continued on next page]

(54) Title: AUTOMATED CINEMATOGRAPHIC EDITING TOOL

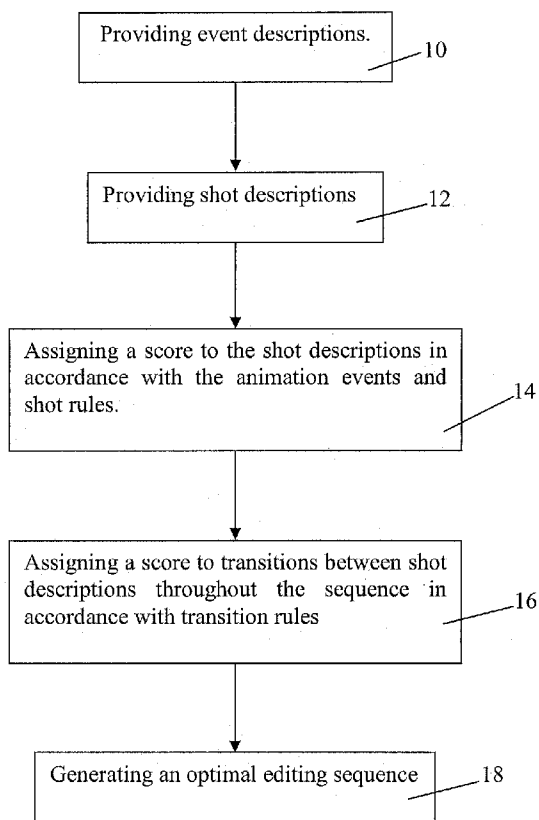


FIGURE 1

(57) Abstract: There is described a method for editing an animation sequence, the method comprising: providing animation events; providing shot descriptions corresponding to shots taken by a plurality of cameras; assigning a score to each one of the shot descriptions in accordance with a corresponding one of the animation events and shot rules; assigning a score to transitions between the shot descriptions throughout the sequence in accordance with transition rules; and generating an optimal editing sequence using the score for the shot descriptions and the score for the transitions between the shot descriptions.

WO 2009/055929 A1



NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG,
CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

AUTOMATED CINEMATOGRAPHIC EDITING TOOLCROSS-REFERENCE TO RELATED APPLICATIONS

[001] The present application claims priority under 35
USC§ 119(e) of Provisional Patent Application bearing serial
5 number 61/001,109, filed on October 31, 2007, the contents of
which are hereby incorporated by reference.

TECHNICAL FIELD

[002] The present invention relates to the field of
cinematography and more specifically, to the use of software
10 to create animations using multiple cameras in an automated
fashion.

BACKGROUND

[003] Traditional methods for creating a fully edited
sequence of film/animation involve a laborious process of
15 generating multiple takes from each of the camera viewpoints,
exporting all the cumbersome video files to a non-linear
editing tool, and manually editing the sequence.

[004] Various software tools are available to help
automate the process. For example, some software will attempt
20 to find optimal camera placements and editings for a given
action sequence by inferring the goals of the animator. This
is a hard and unresolved problem. Other types of software
will attempt to recognize typical situations in the script
and provide standard camera placements for those situations,
25 based on film idioms extracted from cinematographic editing
text-books. The user will be limited to work with these
cameras. This technique imposes constraints by requiring that

given camera positions and orientations are used. By definition, such systems are limited to a finite vocabulary of actions and camera setups, which prevents them from being used in practice. Some techniques alleviate the difficulty of
5 planning a complete editing of a scene by making heuristic editing decisions at runtime, on a frame-by-frame basis, and using finite-state machines switching between cameras automatically, for example. But such systems are unable to plan ahead or back-track. As a result, they are only useful
10 in an interactive or gaming situation, but are ill-suited to a scripted, movie-making situation because they cannot guarantee the level of quality expected in such applications.

[005] Therefore, there is a need to improve the available tools used to create animation sequences.

15 SUMMARY

[006] A movie is composed of one or more scenes and each scene is a piece of continuous action. During production, a scene may be filmed from a variety of viewpoints, by using multiple cameras or by repeating the action with the camera
20 at different locations. Each recording is continuous, and may also be called a take. A take may be composed of one or more shots. A shot may be composed of one or more frames. A frame is a smallest unit of recording. A shot may be obtained by cutting the take at a "mark-in" frame (start frame for the
25 shot relative to the entire camera take) and a "mark-out" frame (end frame for the shot relative to the entire camera take).

[007] Cinematographic editing is the process of cutting and pasting together shots into a single film sequence.

Automatic cinematographic editing is referred to as a method for determining where to cut the shots and how to assemble them into a single scene in an automated fashion.

[008] In accordance with a first broad aspect, there is provided herewith a method for editing an animation sequence, the method comprising: providing animation events; providing shot descriptions corresponding to shots taken by a plurality of cameras; assigning a score to each one of the shot descriptions in accordance with a corresponding one of the animation events and shot rules; assigning a score to transitions between the shot descriptions throughout the sequence in accordance with transition rules; and generating an optimal editing sequence using the score for the shot descriptions and the score for the transitions between the shot descriptions.

[009] In accordance with a second broad aspect, there is provided herewith a system for editing an animation sequence, the system comprising: a processor; a memory accessible by said processor and adapted to store: animation events; and shot descriptions corresponding to shots taken by a plurality of cameras; an application coupled to the processor, the application configured for: assigning a score to each one of said shot descriptions according to a corresponding one of said animation events and shot rules; assigning a score to transitions between said shot descriptions throughout said sequence in accordance with transition rules; and generating an optimal editing sequence using said score for said shot descriptions and said score for said transitions between said shot descriptions.

[0010] In accordance with a third broad aspect, there is provided herewith a system for editing an animation sequence, the system comprising: a memory adapted to store: animation events; and shot descriptions corresponding to shots taken by a plurality of cameras; a shot scoring module connected to said memory and adapted to assign a score to each one of said shot descriptions in accordance with a corresponding one of said animation events and shot rules; a transition scoring module connected to said memory and adapted to assign a score to transitions between said shot descriptions throughout said sequence in accordance with transition rules; and an editing sequence generator connected to said shot scoring module and said transition scoring module, said editing sequence generator being adapted to generate an optimal editing sequence using said score for said shot descriptions and said score for said transitions.

[0011] The method and system may be used such that a scripted list of events described in general terms and a list of available camera choices are read, and an editing sequence (choice of cameras over time) that best translates the scripted events visually are output in accordance with classical continuity style of editing. This has applications in pre-production, production, and post-production of 3D animated movies, live action films of scripted events such as staged theatre productions, and cinematic replay in video games.

[0012] The present method and system are not dependent on a fixed vocabulary of stereotyped actions or camera placements. Instead, they offer a principled and generic framework for evaluating the quality of a shot sequence for

an arbitrarily complex combination of static or dynamic actors and cameras and for finding a non-unique shot sequence with maximum quality in polynomial time.

[0013] In this specification, the term "element" when referring to an action sequence is intended to mean either an actor or an object that is displaced over the course of the action sequence. A shot description is a symbolic description of the content of each frame comprised in the shot, as seen through the lens of the camera. The shot description can be expressed in formal shot description language (SDL).

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] Further features and advantages of the present invention will become apparent from the following detailed description, taken in combination with the appended drawings, in which:

[0015] Fig. 1 is a flow chart of a method for editing an animation sequence, in accordance with an embodiment;

[0016] Fig. 2 is a flow chart of a further method for editing an animation sequence using user constraints, in accordance with an embodiment; and

[0017] Fig. 3 is a block diagram illustrating a system for editing an animation sequence, in accordance with an embodiment.

[0018] It will be noted that throughout the appended drawings, like features are identified by like reference numerals.

DETAILED DESCRIPTION

[0019] There is described herein a method that can be used in a 3D animation tool and/or game engine to automatically generate an "edit list" for editing multiple camera takes into a single edited sequence. This allows animators to quickly get a sense of how their animation work can be pieced together, using the classical continuity style of editing. The edited sequence may be generated and viewed quickly within the animation environment. The method and system may also be used to automatically edit any number of "takes" from several synchronized digital cameras filming a live performance of a scripted scene as long as the individual takes are annotated with the precise screen projections of all actor movements in all cameras.

[0020] The system and method uses camera positions which can be decided by the user and can generate fully edited sequences that are "optimal" with respect to a script of actions and dialogs to be performed by the actors. The output can be used to produce a rough cut of the animated sequences that use the available cameras in a way that best shows the speakers and their bodily actions while keeping a given editing rhythm and observing classical rules of continuity editing, if that is what is desired by the user.

[0021] Figure 1 illustrates an embodiment of a method for editing an animation sequence. Providing animation events is the first step 10 of the method. An animation event is a description of an action which occurs in the animation. For example, an animation event may be defined by a start time, a duration, an action or action class and roles. The roles of an animation event are an ordered list of all the entities

involved in the action corresponding to the animation event. The order of this list may indicate the relative importance of the entities.

[0022] Providing shot descriptions is the second step 12
5 of the method. The shot descriptions describe all possible
shots for creating the editing sequence. A shot description
contains information needed for creating a corresponding
shot. In one embodiment, a shot description can comprise a
10 start time and a duration for the shot, an identification of
the camera used to film the shot and a shot type. The
identification of the camera can be a number. Alternatively,
the identification is provided by the position and
orientation of the camera. The shot type defines the
cinematographic style of the shot. In one embodiment, long
15 shot, medium shot, and close-up represent possible shot
types. Other choices can also be possible such as: extreme
long shot, aerial shot, bird's eye shot, crane shot, over-
the-shoulder shot, one-shot, two-shot, and the like. With one
actor on-screen, the shot is a one-shot. With two actors on-
20 screen facing the camera, the shot is a two-shot. With two
actors on-screen, one facing the camera and one turning away
from the camera, the shot is an over-the-shoulder if that
second actor is closer than the first one. In another
embodiment, the shot type is represented by shot parameters
25 which are defined by elements such as the camera height, pan
and tilt angles, the camera lens, and the like. In this case,
a more varied set of framings or shots is obtained. It
results in a larger number of possible shots from any given
camera position. Furthermore, the shot type may also indicate
30 if the camera is static or dynamic during the shot, and the

trajectory of the camera in the event of a dynamic camera. In another embodiment, a shot description comprises an identification of a take, a "mark-in" frame, and a "mark-out" frame.

5 [0023] The third step 14 of the method corresponds to the assignment of a score to each shot description in accordance with the corresponding animation event and shot rules. The parameters of the shot descriptions are analyzed in accordance with the parameters of the corresponding animation
10 event and the shot rules and the score is assigned as a function of this analysis. For example, a shot description receiving the highest score is the shot description which best represents its corresponding animation event in accordance with the shot rules. Editing style and
15 cinematographic style considerations can be used to create shot rules. Editing style preferences comprise preferences over shot durations and transitions, for example. The style editing preferences may be generic. Alternatively, they can express preferences per action. For example, actions such as
20 talking or smiling may be preferably associated with a close-up shot. Cinematographic preferences are preferences such as those over the shot size, camera angles, and the like. Cinematographic preferences may be generic or action-specific. The score assigned to the shot descriptions
25 provides a ranking of the shot descriptions for each instant of the animation sequence. An instant may be defined as a frame, or may be defined as a given amount of time, such as 10 seconds, 1 minute, etc.

[0024] In one embodiment, the assignment of a score to the
30 shot descriptions is performed as a function of the starting

time and the duration of the shots. Shot durations are explicitly taken into account as part of the ranking of shots and transitions. The duration of shot S_i starting at time t_i followed by shot S_{i+1} starting at time t_{i+1} can be written as

5 $\Delta t_i = t_{i+1} - t_i$. This preference for any given shot is expressed as a function of its starting time t_i and duration Δt_i . This may include preferences over "average shot durations" as a function of shot sizes. Note that this allows

10 all possible edits points between any two shots to be evaluated.

[0025] In one embodiment, a score is assigned to each frame comprised in the shot corresponding to the shot description, in accordance with the corresponding animation event and shot rules, and the score assigned to the shot

15 description is calculated in accordance with the score assigned to each frame.

[0026] The fourth step 16 of the method is the assignment of a score to the transitions between shot descriptions according to transition rules. Editing style and

20 cinematographic style considerations can be used to create the transition rules.

[0027] Some examples of transition rules are:

(1) cuts between views of the same actor must keep the actor in the same side of the screen, with the same motion

25 direction, and with a significant difference in either size or profile;

(2) cuts between views of the same actors must maintain left/right relationships; and

(3) cuts from one actor to another must maintain the proper distance between them

[0028] In one embodiment, transitions are scored according to how well they support the classical continuity editing style, which we summarize here:

- Line of action. The relative ordering of characters must remain the same in the two shots.
- Screen continuity. Characters who appear in both shots should not appear to jump around too much.
- 10 • Looks. The gaze directions of characters seen in separation should match. If they are looking at each other, their images should also be looking at each other. If the two characters are NOT looking at each other, their images should NOT be looking at each other
- 15 • Distance between characters. The sum of apparent distances to two characters shown in separation should be at least twice the actual distance between them (as if the two images were taken from the same camera position). This prevents the use of close-ups for two
- 20 characters very far apart.
- The shot size relative to a character should change smoothly, rather than abruptly, thus avoiding cutting from a long shot directly to a close-up. Instead, we would prefer to first cut to a medium-shot, then to a
- 25 close-shot.

- Motivation. Cutting during an action is usually more pleasant. In that case, a lower transition cost can be used, and motion continuity is considered.
- Focus of interest. A character with more screen space is probably more important in evaluating the transition.

5
10 [0029] In one embodiment, at least some of the shot and transition rules correspond to preferences which are input by a user of the method. In this case, the method further comprises a step of receiving preferences about the editing style and cinematographic style from the user.

15 [0030] In another example, the user inputs an exemplary edited sequence representative of a given editing and cinematographic style. The exemplary edited sequence comprises at least one action script, a set of exemplary shot descriptions, and an exemplary edit decision list. Editing and cinematographic style parameters which represent the user preferences over editing and cinematographic style are determined from the exemplary edited sequence. The shot rules and the transition rules are then generated in accordance
20 with the editing and cinematographic style parameters.

25 [0031] It should be understood that a score may be a mark such that the shot/transition having the highest score/mark corresponds to the shot/transition presenting the highest quality. Alternatively, the score may correspond to a cost, i.e. the shot/transition having the highest score/cost corresponds to the shot/transition presenting the lowest quality.

[0032] In one embodiment, a cost is first assigned to a shot and a score is subsequently calculated from the assigned cost. A cost is an additive, positive value that measures violation of the rules of editing, and the value of a cost is within $[0; +\infty[$. Scores and preferences are multiplicative, positive values between 0 and 1. A score can be derived from a cost as being the exponential of minus the corresponding cost function: $0 < \text{score} = \exp(-\text{cost}) < 1$.

[0033] When preferences are received from a user, a cost function can be evaluated as being minus the log of the corresponding preference or score.

[0034] Furthermore, probabilities can be derived from scores and preferences via normalization with a normalization factor $Z = \text{sum score over all possible output shot sequences}$, given the same input action sequence. Probabilities are normalized, multiplicative, and positive values comprised between 0 and 1. A sequence probability is defined as a conditional probability for an output shot sequence, given an input action sequence and a set of style parameters. In other words, a sequence probability represents the probability of choosing a shot sequence for a particular action sequence, given the style parameters.

[0035] While it may be needed for comparing shot sequences with different actions or styles, for any given input action sequence, normalization can be omitted during the step of searching for the best output sequence since all sequences share the same normalization factor Z .

[0036] The last step 18 of the method is the generation of the optimal editing sequence. An editing sequence is a time-

ordered list of shot descriptions. All possible editing sequences are generated and a score is assigned to each editing sequence as a function of the scores of the shots and transition comprised in the editing sequence. The optimal editing sequence is the editing sequence which has the highest mark or the lowest cost. In other words, the optimal editing sequence identifies the shot type and the camera to be used at each instant in time of the sequence.

[0037] In one embodiment, the optimal editing sequence is returned to the user under the form of an edit decision list (EDL) which is a representation of the result of cinematographic editing. The edit decision list can be in the form of a sequence of shots, each shot being identified by a camera take, a "mark-in" frame and a "mark-out" frame. For example,

```
shot1 = (take 10, markin=50, markout=100);
```

```
shot2 = (take 11, markin=10, markout = 20)
```

is an EDL for a 60 frame-long movie where the first shot uses frames 50 through 100 of take 10, and the second shot uses frames 10 through 20 of take 11.

[0038] In one embodiment, the EDL can be further elaborated to allow for gradual transitions such as dissolves, superimpositions, fades and special effects such as frame-rate changes, etc.

[0039] In one embodiment, the optimal editing sequence is sent to the animation generator which creates the animation shots according to the shot descriptions of the optimal editing sequence. In one embodiment, the optimal editing

sequence is made available to the user prior to the generation of the animation. For example, the first frame of each shot may be displayed on the user's interface to provide the user with a first impression of the animation. The displayed editing sequence corresponds to a proposal and the user may be asked to validate the displayed editing list. If he rejects the editing list, the user may be presented with a second editing list which corresponds to the second optimal editing list, i.e. the editing sequence having the second highest mark or the second lowest cost. Alternatively, the user may be asked to input new preferences. The user may also be provided with the complete list of the possible editing sequences and their respective score, from most optimal to least optimal.

[0040] In one embodiment, the method illustrated in figure 1 further comprises the step of generating the shot descriptions. The shot descriptions are generated using the received inputs, namely the set of positions and orientations for the cameras, the layout of a scene for the animation sequence comprising decor and elements therein, the trajectory of the elements in the animation sequence, and the animation events.

[0041] In another embodiment, the shots of the scenes are created before the generation of the optimal editing sequence. In this case, the shot descriptions are generated in accordance with the shots. For example, a TV show may be filmed by multiple cameras resulting in numerous shots. A shot description is created for each shot and these shot descriptions are used as input in the method for generating the optimal editing sequence. The editing of the TV show is

then performed in accordance with the generated optimal editing sequence of shots.

[0042] In one embodiment, the method further comprises the step of receiving a script. A script is a symbolic description of all actions taking place in the scene. The following presents a simple example of a script:

Event (type=ASK, agent=A, dest=B, start=T1, end=T2)

Event (type=ANSWER, agent=B, dest=A, start=T3, end=T4)

Event (type=EXIT, agent= B, start=T5, end=T6)

[0043] The layout, the trajectories, the animation events and the shot descriptions are determined directly from the script. Any method known by a person skilled in the art to extract this information from the script may be used. For example, the extraction step may comprise the generation of hierarchical state machines for all actions present in the script.

[0044] In one embodiment, the method further comprises the step of generating additional camera positions and orientations, aimed at the actors or objects of interest in the scene. Such cameras can then be used in the evaluation of the best editing sequence. This step can be executed using a variety of available methods, including «through-the-lens» camera control, stereotyped camera placements taken from classical editing textbooks, and the like. This additional step makes it possible to use the method even when the user of the system does not provide any camera choices.

[0045] This section presents an example for illustration purpose. The input comprise three actions:

- event 1: actor A asks a question to actor B;
- event 2: actor B answers the question; and
- 5 - event 3: actor B leaves the scene.

[0046] Three cameras are available to create the animation: camera 1 being static and directed towards A from behind B's shoulder, camera 2 being initially directed towards B from behind A's shoulder, and then panning with B
 10 as B leaves the scene, and camera 3 being static and filming a general view of the two actors. This results in several shot choices for every action, namely MS A OTS B, MS B OTS A, PAN with B, LS A and B. Omitting shot choices with very low scores, the following table illustrates the received shot
 15 descriptions and their corresponding scores:

Event A ($t_1 \rightarrow t_2$)	Event B ($t_2 \rightarrow t_3$)	Event C ($t_3 \rightarrow t_4$)
Shot 1: MS A over B's shoulder with camera 1 (80%)	Shot 3: MS B over A's shoulder (80%) with camera 2	Shot 5: PAN with actor B (80%) with camera 2
Shot 2: LS A and B with camera 3 (50%)	Shot 4: LS And B with camera 3 (50%)	Shot 6: LS A and B with camera 3 (50%)

Table 1: Possible shots and their associated scores, before user preferences applied.

[0047] One preference entered by a user is to have long shots. The preference of the user for long shots is

represented by multiplying the previous shot scores with 100% for long shots and 50% for other shot sizes:

Event A ($t_1 \rightarrow t_2$)	Event B ($t_2 \rightarrow t_3$)	Event C ($t_3 \rightarrow t_4$)
Shot 1: MS A over B's shoulder with camera 1 (40%)	Shot 3: MS B over A's shoulder (40%) with camera 2	Shot 5: PAN with actor B (40%) with camera 2
Shot 2: LS A and B with camera 3 (50%)	Shot 4: LS A and B with camera 3 (50%)	Shot 6: LS A and B with camera 3 (50%)

Table 2: Possible shots and their associated scores, after user preferences applied.

- 5 [0048] Transitions between shots are computed separately based on the relative screen positions of actors A and B before and after the transitions. By definition, a transition score of 100% is assigned when the same camera is used for two following shots.

Transition	Score
Shot 1 → Shot 3	60%
Shot 1 → Shot 4	30%
Shot 2 → Shot 3	60%
Shot 2 → Shot 4	100%
Shot 3 → Shot 5	60%
Shot 3 → Shot 6	50%

15277501

Shot 4 → Shot 5	20%
Shot 4 → Shot 6	100%

Table 3: Transitions and their associated scores

[0049] Multiplying the scores for the individual shots and the transitions, it is possible to assign a score to each possible sequence, as illustrated in table 4:

Sequence	Score
Shot 1 → Shot 3 → Shot 5	18.4%
Shot 1 → Shot 3 → Shot 6	9.6%
Shot 1 → Shot 4 → Shot 5	1.9%
Shot 1 → Shot 4 → Shot 6	6.0%
Shot 2 → Shot 3 → Shot 5	11.5%
Shot 2 → Shot 3 → Shot 6	6.0%
Shot 2 → Shot 4 → Shot 5	4.0%
Shot 2 → Shot 4 → Shot 6	12.5%

5 Table 4: Possible sequences and their associated scores, before user preferences applied.

Sequence	Score
Shot 1 → Shot 3 → Shot 5	2.3%
Shot 1 → Shot 3 → Shot 6	2.4%

Shot 1 → Shot 4 → Shot 5	0.5%
Shot 1 → Shot 4 → Shot 6	3.0%
Shot 2 → Shot 3 → Shot 5	2.8%
Shot 2 → Shot 3 → Shot 6	2.9%
Shot 2 → Shot 4 → Shot 5	2.0%
Shot 2 → Shot 4 → Shot 6	12.5%

Table 5: Possible sequences and their associated scores, after user preferences applied.

[0050] From table 4, the optimal sequence is: Shot 1 → Shot 3 → Shot 5, which has the highest score before assigning preferences to long shots. From table 5, the optimal sequence is: Shot 2 → Shot 4 → Shot 6, which is in fact a single long shot of actors A and B, and has the highest score after user assigned preference to long shots.

[0051] It should be understood that this example is an extreme simplification of which the purpose is to illustrate the principle of the above described method. The scores presented above are arbitrary and do not represent an actual simulation. In real situations, shot scores and preferences are evaluated at regular time intervals, using a combination of all actions occurring during each interval, and preferences over shots durations are also taken into account.

[0052] It should be understood that other methods for assigning a score to a sequence may be used. For example, a weighting factor can be assigned to the shot score and the transition score so that the transitions are more important

than the shots in the calculation of a sequence score, or vice versa. In another embodiment, the scores for the shots and the transitions may be added instead of being multiplied.

[0053] For the purposes of the following description, the following notations are used:

K is the number of cameras.

M is the number of shot classes (number of cameras, each counted multiple times to account for different lens and angle choices).

I is the sampling time interval (in seconds). For example, 2 seconds for a rough cut, $1/24^{\text{th}}$ of a second for a final cut.

N is the number of time intervals (duration of entire scene in seconds is $N \times I$).

P is the number of shots in the final edited scene.

t is the current time interval (in unit of I, so that the actual time is $t \times I$ in seconds).

D is the maximum duration of a shot.

$X_k(t)$ is the trajectory in time for entity k.

E_k is the animation event associated with action m. E_k is defined by a starting time, a duration, an action, and roles associated with the action, so that: $E_k = (\text{start}_k, \text{dur}_k, a_k, r_k)$.

$S_i = (t_i, \Delta t_i, c_i, s_i)$ is the description of a shot i starting at time t_i , having a duration Δt_i , taken by camera i and having shot parameters i. Another notation is also used for the shot

descriptions, namely $S_i(t)$. In this case, $S_i(t)$ is expressed as:

$$S_i(t) = \begin{cases} (c_i, s_i) & \text{for } t_i \leq t < t_i + \Delta t_i \\ (0, 0) & \text{for } (t < t_i) \text{ and } (t \geq t_i + \Delta t_i) \end{cases}$$

5 $A_{mn}(t)$ is the transition cost from shot m starting at time $t-1$ to shot n starting at time t .

$B_m(t)$ is the cost function for S_m at time t .

[0054] Figure 2 illustrates an embodiment of a method for editing an animation sequence using additional user
10 constraints. The first step 20 is the reception of inputs. The inputs comprise a set of positions and orientations $C_m(t)$ for each one of a plurality of cameras involved in shooting an animation sequence, a layout for the animation sequence, including decor and elements therein, the trajectory $X_k(t)$ of
15 the elements in the animation sequence, and animation events $E_k(\text{start}, \text{duration}, \text{action}, \text{roles})$. The layout for the animation sequence describes where each item making up the scene is positioned. This includes static items, such as furniture, walls, decoration, etc, and dynamic items, such as
20 elements that may move during the action sequence. The trajectories of dynamic elements describe how the elements move and where they end up.

[0055] In one embodiment, the trajectories of the elements are fixed or input by a user. In another embodiment, the
25 trajectories are determined from the animation events. Any method known by a person skilled in the art to generate the trajectories from the animation events can be used. For example, a combination of methods such as path-planning,

collision avoidance, automatically-generated hierarchical state machines, and searching a motion graph through a library of existing movements can be used to determine the trajectories.

5 [0056] In one embodiment, the camera positions and orientations are defined by a user of the method. For example, the user may select the camera positions and orientations from a list of randomly generated camera positions and orientations. This input provides a constraint
10 for the cameras to be used. Alternatively, the selection of the cameras and their associated positions and orientations is performed by the system which generates the editing sequence.

[0057] The extraction of Shot Descriptions $S_m(t)$ using the
15 received inputs constitutes the second step 22 of the method. These shot descriptions are then used as input, in addition to editing style and cinematographic style considerations, to determine Shot Transition Costs $A_{mn}(t)$ and $B_m(t)$, referred as steps 24 and 26. The next step 28 is the computation of
20 partial cost function $BEST(m,t)$ which is used to search the entire space of the solutions. Finally, the last step 30 is the generation of the optimal editing sequence $S_0, \Delta t_0, S_1, \Delta t_1, \dots, S_p, \Delta t_p$ compatible with any user constraints $S_u, \Delta t_u$ which can be a particular shot description S_u imposed by the
25 user, for example. Having long shots for the first and the last shots of a scene is another example of a user constraint. It should be understood that the editing style and cinematographic style parameters that are considered are subjective and may vary greatly between applications. In
30 addition, if the shot descriptions have already been

generated, the system may be used to perform only the second part of the process.

[0058] In one embodiment, all of the available shots (world situations x camera positions) are described. In another example, the camera height, pan and tilt angles, and the lens may be changed to obtain a more varied set of framings or shots. It results in a larger number of possible shots from any given camera position.

[0059] The following example is given for a simple case where both cameras and the scene are relatively static. It should be understood that the present method and system is not limited to the static case and the following is exemplary only. In this example, the shot parameters are limited to three choices, namely, one-shot, two-shot, and over-the-shoulder shot.

[0060] Shot sequencing is used to rank the shots with respect to actions performed by the actors. For example, given a script of M actions, the sequence of action events may be expressed as:

$E_1 = (\text{start}_1, \text{dur}_1, a_1, r_1), E_2 = (\text{start}_2, \text{dur}_2, a_2, r_2), \dots, E_M = (\text{start}_M, \text{dur}_M, a_M, r_M)$

A solution is a sequence of P shots:

$S_1 = (t_1, \Delta t_1, c_1, s_1), S_2 = (t_2, \Delta t_2, c_2, s_2), \dots, S_P = (t_P, \Delta t_P, c_P, s_P)$

Cuts occur whenever the camera changes between successive intervals. Reframing occurs when the camera remains the same but the shot description changes. Note that cuts and reframings may occur at times t_1, t_2, \dots, t_P different from the

start times of the actions. The number of possible editings of a given action script is very large. As a result, the process of choosing the best editing is a highly combinatorial problem. The present system provides a solution
5 for computing preferences over all possible editings of M cameras over N time intervals, based on a script of actions and a representation of classical cinematographic and editing styles. Preferences are defined in such a way that the best sequence can be found efficiently (in polynomial space and
10 time) using dynamic programming.

[0061] In one embodiment, the cost of a sequence is computed from at least one of three types of preferences: stylistic preferences on shot types and durations, preferences on which shot types to use for each action in the
15 script (preferred association of shot types with actions or action types), and preferences on how to cut from one shot to the next one. The first two preferences are used to calculate the cost of the shots, while the third preferences are used to calculate the transition costs. Assuming the preference
20 for any given shot, duration or transition is measured by a number in $[0,1]$, we may give the preference a probabilistic interpretation and use the negative logarithm of the preference as the associated cost, so that a preference of 0 has an infinite cost and is never used, while a preference of
25 1 has zero cost. Note that in some cases, we may prefer to assign additive 'scores' in $[Min, Max]$ and derive the cost functions as $Max - score / (Max - Min)$ in $[0,1]$. In those cases, the preferences can be derived from the costs or scores with functions of the form $1/Z \exp(-cost) = \exp(\alpha + \beta * score)$ where Z, α and β are normalization constants.
30

[0062] The objective is to find the shortest path through M cameras pointing at the scene, which means the path having the lowest cost (highest score) possible. With respect to preferences, we get the shot sequence with the largest joint probability, measured by the product of all shot preferences. Note that the best sequence is generally made of P consecutive shots.

[0063] As the number of possible shot sequences to be evaluated may increase dramatically with the number of possible shots, the assignment of shot and transition scores may be performed within a semi-Markov assumption, so that the space of solutions can be searched efficiently and the optimal solution can be found in all cases. The method uses conditional random fields which are discriminative probabilistic models used for the labeling or parsing of sequential data, such as natural language text or biological sequences.

[0064] In one embodiment, we model shot preferences with a semi-Markov model and editing decision lists are modeled as semi-Markov conditional random fields, meaning that the probability or preference for a shot starting at time t_i with a duration of Δt_i is the product of the global preferences for that shot (based on its duration or size) with the preferences over all time intervals between t_i and t_{i+1} . A semi-Markov conditional random field is a particular case of conditional random fields which models variable-length segmentations of a label sequence. Semi-Markov conditional random fields allows to model long-range dependencies, at a reasonable computational cost.

[0065] In one embodiment, the semi-Markov assumption allows the adaptation of the forward-backward and Viterbi algorithms to a conditional random field in order to determine the optimal editing sequence, given the input
5 action script, and training algorithms developed for conditional random fields can be adapted when cinematographic and editing style parameters are learnt directly from examples.

[0066] In terms of costs, the cost associated with the
10 shot is the sum of the costs for all its time intervals t_i plus the cost associated with the duration Δt_i and other style parameters of the shot.

[0067] Concerning the first preferences, namely stylistic preferences on shot types and durations, an example of style
15 parameters is the average shot length which measures the rhythm of the editing. For example, shots whose durations are modeled by a log-normal distribution with an average shot length (ASL) m and a standard deviation σ may be preferred. The cost C associated with a shot of length Δt can then be
20 expressed as:

$$C(\Delta t) = -\log p(\Delta t) = \log \Delta t + (\log \Delta t - \log m)^2 / 2 \sigma^2$$

[0068] Other examples of style parameters are the desired ratios of long shots, medium shots, and close shots. All of these can be translated into preferences over the parameters
25 of the shots, as explained below.

[0069] The editing rhythm may be set by choosing an ASL for the sequence. The rhythm may also be fine-tuned by choosing the variance σ^2 and autocorrelation function R of

the shot lengths. In addition, the perceived length of a shot may depend on its size, its novelty, and the intensity of the action. For the purpose of this example, the ASL is set as a function of shot size, so that close-ups are cut faster and long shots are cut slower. Assume that the ratio of long shots, medium shots, and close shots are set to be N_1 , N_2 , and N_3 . Assume further that the average shot lengths are $m_1 = a_1L$, $m_2 = a_2L$, and $m_3 = a_3L$ where a_1 , a_2 , a_3 are constant factors. Then the general ASL will be:

$$1/N * (N_1 m_1 + N_2 m_2 + N_3 m_3) = L/N * (N_1 a_1 + N_2 a_2 + N_3 a_3)$$

where N is the total number of shots ($N = N_1 + N_2 + N_3$)

and we can set:

$$L = N m / (N_1 a_1 + N_2 a_2 + N_3 a_3)$$

so that the overall ASL for a sequence of N_1 long shots, N_2 medium shots, and N_3 close shots is expected to be "m". For any given shot in any given size, the formula $C(\Delta t)$ can express the costs as a function of duration.

[0070] In a simple example with two characters, each performing a sequence of actions a_t and b_t , shot preferences at any given time are computed as a function of the action categories for a_t and b_t . Action categories that can be used in a dialog situation are speech actions, facial expressions, hand gestures, and pointing gestures. Shot choices are ranked for each category of each character.

[0071] Concerning the second preferences, namely preferences on which shot types to use for each action, these shot preferences may be expressed as a function of a small

number of visual goals that are extracted automatically from the animation events or the script. The actions are categorized into a small number of visual categories, such as speech actions, facial expressions, hand gestures, pointing gestures, and bodily gestures. Each category is associated with a list of natural visual goals such as showing the face, the hands, the entire body, or the environment of the corresponding role. Shot choices are ranked for each category. Then, the ranking of a shot of any given time interval is computed with the product of the preferences for that shot over all actions occurring at that time. Similarly, the cost associated with the shot at any given time is the sum of the costs for all actions occurring at that time.

[0072] Concerning the third preferences, namely preferences on how to cut from one shot to the next one, these preferences are expressed based on film grammar. Examples of third-type preferences are: keep each character on the same side of the screen across a cut, avoid cutting between two-shots, avoid cutting from a long-shot to a close-shot, prefer cutting between shots for the same size (long, medium or close), etc.

[0073] In one embodiment, the shot descriptions, shot preferences and costs are translated into a numerical, sub-symbolic shot description language (SDL) which includes the screen positions and motion vectors of the character's body parts and gaze directions. The rules of film grammar are also translated into SDL. As a result, the shot preferences and the transition preferences between M arbitrary shots can be computed efficiently as a function of their low-level descriptions, as described below.

[0074] This section explains the SDL language used to describe the content of a shot at any given time in screen coordinates. Shots with one actor are described with the character's name (in the scene); the screen coordinates and
5 depth of the middle of the eyes (the depth z is given in units of focal length multiplied by sensor size, so that the head size in screen coordinates is s/z if s is the head size in world coordinates); the onscreen size of the actor's head (excluding offscreen or occluded parts); the profile angle
10 for the actor's head, in degrees (0 if the actor faces the camera, 90 for a left profile, -90 for a right profile, 180 for a back view), relative to a vertical line passing through the eyes; the camera angle relative to the actor's eyes, in degrees (0 for a shot at eye-level, 90 for a top view, -90
15 for a bottom view), defined relative to a horizontal plane passing through the eyes. When actors are turning around, it may become useful to also take note of the profile and camera angles relative to their upper body (using the middle of the shoulders) and lower body (using the middle of the hips).
20 Typical one-shots can be defined using short-hand notations such as ONE-SHOT CU MAN, 34LEFT, LOW ANGLE, SCREEN CENTER which translates to name='MAN', eyes=(0, 2/3, 1), profile=45, angle=-30 in SDL. As another example, ONE-SHOT LS WOMAN, 34BACK RIGHT, SCREEN LEFT translates to name='WOMAN',
25 eyes=(2/3, 2/3, 12), profile=-135, angle=0, etc.

[0075] Shots with two or more actors can be described with separate descriptions for all actors. In addition, in the case of two actors, we describe two-shots with their own vocabulary: the two character's names (in the scene); the
30 screen coordinates and depth of the middle of the eyes for

the two actors, even when they are off-screen. This gives us the line of action between them; the onscreen sizes of their heads (excluding off-screen or occluded parts); the profile angle relative to the line of action, in degrees (0 if the actor faces the camera, 90 for a left profile, -90 for a right profile, 180 for a back view), defined relative to a vertical line passing through the center of the two actor's eyes; the camera angle relative to a line of action, in degrees (0 for a shot at eye-level, 90 for a top view, -90 for a bottom view), defined relative to a horizontal plane passing through the center of the two actor's eyes. Again, typical two-shots can be defined using short-hand notations. For instance, TWO-SHOT LS MAN WOMAN translates in SDL to

5 name[0]='man', name[1]='woman', eyes[0]=(2/3,2/3,12),

10 eyes[1]=(1/3,2/3,12), profile=90, angle=0. TWO-SHOT LS WOMAN

15 MAN translates to name[0]='woman', name[1]='man',

 eyes[0]=(2/3,2/3,12), eyes[1]=(1/3,2/3,12), profile=90,

 angle=0. TWO-SHOT CS MAN OTS WOMAN translates to

20 name[0]='man', name[1]='woman', eyes[0]=(2/3,2/3,4),

 eyes[1]=(1/3,2/3,3), profile=15, angle=0.

[0076] SDL also describes the dynamics of shots over a time interval $I(t)$ by storing the time-derivatives (motions) of all its parameters. For longer time intervals $I(t)$ of ten movie frames or more, the minimum and maximum values of all

25 parameters can also be stored.

[0077] This section explains the process of describing a shot at a given time, given the 3D scene and the camera parameters at that time. Let C be the position of the camera's optical center, V the direction of its optical axis,

30 F the focal length of the camera lens, H and V the size of

the image sensor. Perspective projection of a scene point P in (3D) world coordinates to an image point p in (2D) projective camera coordinates is written $p = K(Rc P + Tc)$ where R is a 3X3 rotation matrix which depends on C and V , T a translation vector equal to $-RC$ and K a 3X3 matrix of the internal parameters F, H and V . Similarly, the projection of a vector V is a vector $v = KRP$.

[0078] Symbolic projection of a 3D scene into M cameras comprises the following steps:

- 10 (1) Compute bounding boxes of all scene elements, including all actor's faces, upper bodies and lower bodies, in their own 'object' coordinates at time $t = 0$

Then, for all times t' in $I(t)$, for all actors in the scene and for all cameras

- 15 (2) Retrieve center $C_k(t')$ and principal directions $UP(k,t')$, $DOWN(k,t')$, $LEFT(k,t')$, $RIGHT(k,t')$, $FRONT(k,t')$ and $BACK(k,t')$ of element and project them into camera coordinates. This gives the bounding boxes of all elements in projective coordinates, including depth.

- 20 (3) Sort all elements according to depth and resolve visibility. Compute SDL description for all visible actors, including angles between V and $FRONT(k,t')$ for all actors.

- (4) Store $[min,max]$ values of all SDL descriptors and their time derivatives into shot description $S(t)$. Many variations are possible for efficiently applying this step as a function of the time interval size.
- 25

[0079] In the special case of two-shots, additional steps are taken to compute the vector between the two actors (line of action). The same basic steps can be used to compute other optional elements in the SDL following the same line.

5 [0080] This section describes the process of decomposing the animation script into a list of actions occurring at any time. The input in this step may differ from one embodiment to another, with respect to e.g. the vocabulary and representation of actions. For instance, actions may be
10 restricted to simple subject-verb-object triplets, or even to a small list of one-person and two-person actions. To be general, we assume a representation of actions as 'animation events' that comprise the start time and the duration of the event; the action class of the event (such as speech, facial-
15 animation, pointing, looking, etc.) and its roles (such as agent, object, patient, instrument, etc.). The roles of an action are a numbered list of scene elements participating in the action. As a simple illustration, monologue actions typically involve a single 'agent' role (which may be
20 implied, as in speak(agent:man) while dialog actions typically involve one 'agent' role and one 'patient' role (as in speak(agent: man, patient: woman). We use a lexicon of action classes indicating which parts of the different roles (typically, the face, lower body or upper body) are important
25 for that particular action. In practice, it is sufficient to categorize actions into a small number of classes, such as facial actions, hand gestures and displacements. Note that the class labels can also be partially recovered from the associated animation curves by using motion analysis, in
30 cases where the action script does not include them.

[0081] The analysis of the action script consists of two main steps:

- (1) For all actions in the script, we build the list of scene elements given by all the action's roles; and
- 5 (2) For all time intervals t , we build the list of actions that are taking place at that time.

[0082] This section explains the process of computing the cost of a shot over a time interval, given its SDL description S_i and the list of actions occurring in that interval. Note that this is expressed with respect to the chosen cinematographic style. Given the action decomposition of the previous section, a very general strategy for choosing the shots is to assign screen space to actions and their roles according to their importance in the script. We can
 15 measure the amount of screen space devoted to an event E_k as

$$\text{sum}(t) \text{sum}(r \text{ in } E_k) \text{screensize}(r \text{ in } S_i)$$

[0083] Let 'occur(E_k, t)' be a boolean function that indicates that animation event E_k is taking place at time t . A common criterion known as the «Hitchcock principle» states
 20 that the best shot is the one that maximizes the correlation between the screen sizes of all actors and their importance in the script at that particular time. This correlation can be approximated as follows:

$$\text{sum}(E_k) \text{occur}(E_k, t) \text{sum}(r \text{ in } E_k) \text{screensize}(r \text{ in } S_i) \\
 25 \quad \times \text{weight}(r \text{ in } E_k) \times \text{weight}(E_k \text{ in story})$$

[0084] This provides a good measure of how well the shots represent the action. The two 'weight' functions measure the

importance of the roles in the actions and the importance of the actions in the story. The first 'weight' function is typically built into the action lexicon. The second 'weight' function can be read from the script or annotated by the user. In other cases, a default value of 1.0 is used for all roles and actions.

[0085] In one embodiment, the 'screensize' function is computed from the SDL shot description as:

$$2 \times \text{onscreen}(\text{object}) / (\text{imagesize} + \text{size}(\text{object}))$$

[0086] Recall that $\text{onscreen}(\text{object})$ in S_i measures the image size of the visible part of the object while $\text{size}(\text{object})$ measures the total object size (including off-screen and occluded parts). The associated cost functions are:

$\text{screencost}(\text{object}) = 1 - \text{screensize}(\text{object})$ and

$$\text{actioncost}(S_i, t) = \sum (E_k) \text{ occur}(E_k, t) \sum (r \text{ in } E_k) \text{screencost}(r \text{ in } S_i) \times \text{weight}(r) \times \text{weight}(E_k)$$

which results in a score of $\exp\{-\text{actioncost}(S_i, t)\}$.

[0087] This provides the basis for a general shot ranking algorithm. Details are omitted here on how we can also take into account the screen positions and angles of actors, relative to the camera, or other scene elements in the background. Many changes or additions can be made to the general approach. In all cases, rankings can efficiently be computed from the stored shot descriptions S_i and stored into a single table $B(i, t)$.

[0088] This section explains the process of computing the cost of a transition between two shots, given their SDL descriptions. Transition between shots are ranked by comparing the screen sizes, locations, and angles for all
5 actors in the SDL descriptions of the two shots. Note that this is expressed with respect to the chosen editing style.

[0089] In one embodiment, transition rules are translated into the following cost functions

- *screensize(actor,shot)* measures how much of the actor
10 is visible in the shot.

- *jumpmotion(actor,shot,shotj)* penalizes jumps in motion directions.

- *jumpscreen(actor,shot,shot)* penalizes jumps in screen positions.

15 - *jumpprofilesize(actor,shot,shot)* penalizes shots with the same profile and size.

- *jumpleftright(actor,actor,shot,shot)* penalizes inconsistent left/right relations between actors in the two shots.

20 - *distance(actor,actor,shot,shot)* keeps the apparent distance between actors consistent with their actual distance in the animation.

[0090] Those are numerical functions of the SDL descriptions of the two shots. They require scaling
25 parameters, which are chosen carefully. Scaling parameters measure the importance given to the corresponding rules.

Scaling parameters are a function of editing style and can be learned from examples of that style, using expert knowledge and/or statistical methods. It is possible to deliberately break a rule by setting its scaling parameters to zero or even negative values.

[0091] Based on the above cost functions, the following algorithm can be used for computing transition costs:

```
actioncost = 0

for all actors ak who appear in both shots

10   compute wk = min (i=1,2) screensize(ak,si)

      compute jumpk = jumpscreen(ak,s1,s2)

          + jumpprofilesize(ak,s1,s2)

          + jumpmotion(ak,s1,s2)

      actioncost += wk*jumpk

15  for all actors ak,al who appear in both shots

      compute wk = min (i=1,2) screensize(ak,si)

      compute wl = min (i=1,2) screensize(al,si)

      if ak and al have different left/right relations

          actioncost += wk*wl*jumpleftright(ak,al,s1,s2)

20  for all actors ak1, ak2 who appear separately

      compute w = min(i=1,2) screensize(aki,si)

      actioncost += w*distance(ak1,ak2,s1,s2)
```

15277501

[0092] The above algorithm provides the basis for a general cut ranking algorithm. The rankings can efficiently be computed 'on-the-fly' from shot descriptions S_i and S_j of arbitrary cameras, and can be stored into a single table $A(i,j,t)$. In addition, it is also possible to take into account the gaze directions as follows:

for all actors ak_1, ak_2 who appear separately

compute $w = \min(i=1,2) \text{ screensize}(aki, Si)$

if ak_1 is looking at ak_2 in the world but not in the shot

10 *actioncost += w*look*

if ak_1 is looking at ak_2 in the shot but not in the world

*actioncost += w*look*

[0093] More generally, we can translate film grammar rules in a systematic fashion if the rules are written in terms of typical shot descriptions. Positive rules describe which shots should follow which shots. Negative rules describe which shots should not follow which shots. Preference rules tell which shots should be preferred after which shots. All such rules translate naturally into cost functions similar to the examples presented above.

[0094] This section explains how we enumerate all possible editing sequences, which includes ranking all possible transitions between any two shots at any edit point. This is a combinatorial problem which we resolve with a dynamic programming solution, based on the observation that our ranking function has the semi-Markov property. Indeed, our

ranking function has the property that the cost of an edit decision list $S_1=(t_1, \Delta t_1, c_1, s_1), S_2=(t_2, \Delta t_2, c_2, s_2), \dots, S_P=(t_P, \Delta t_P, c_P, s_P)$ can be decomposed as follows:

$$\begin{aligned}
 L[S_1=(t_1, \Delta t_1, c_1, s_1), S_2=(t_2, \Delta t_2, c_2, s_2), \dots, S_P=(t_P, \Delta t_P, c_P, s_P)] &= LS(S_1, t_1, t_2) + LT(S_1, S_2) \\
 &+ LS(S_2, t_2, t_3) + LT(S_2, S_3) \\
 &+ \dots \\
 &+ LS(S_{n-1}, t_{n-1}, t_n) + LT(S_{n-1}, S_n) \\
 &+ LS(S_n, t_n, t_{n+1})
 \end{aligned}$$

10 where $LS(S_i, t_i, t_{i+1})$ is the cost associated with shot $S_i=(t_i, \Delta t_i, c_i, s_i)$ between t_i and t_{i+1} and $LT(S_i, S_j)$ is the cost associated with a transition from shot S_i to shot S_j .

[0095] As a result, the total cost function has the property that the shortest path that includes a given shot S_i ending exactly at time t_{i+1} must necessarily be built from the shortest path leading to that shot and the shortest path starting at that shot. This is the basis for an efficient dynamic programming algorithm which computes and stores the best partial path leading to shot S_i ending exactly at time t using a simple recurrence relation. Let $LM(S_i, t)$ be the cost of that partial path. Then the partial path associated with shot S_i ending exactly at time $t' > t$ can be found by searching for the minimum of:

$$LM(S_i, t) + LS(S_j, t+1, t') + LT(S_i, S_j)$$

25 over all possible choices of i and $t < t'$. With M shot types and a maximum shot duration D , this requires the evaluation

of $D(M-1)$ solutions. As a result, we can find the optimal edit list for a sequence with N time intervals with only NM^2D operations and a storage space of NM .

[0096] For action-shot preferences, one embodiment
5 comprises building a table of the best shot to be used for each action. For example, a dialog action can be portrayed by showing the speaker or the reaction of the other actor, or both. A facial expression can be portrayed by showing the actor's face, the closer the better. A hand-gesture can be
10 portrayed by showing the actor's hands and face. A close-up of the hands would usually not work because it would hide which actor is actually gesturing. A pointing gesture should show the pointing actor's hand and head as well as the object, actor, or place being pointed at, either at the same
15 time or in succession. We use such input to recompute the cost of any given shot S_m as the sum of distances between S_m and the best shots S_k associated with all actions A_k occurring in the time interval.

[0097] Exemplary code for computing an optimal sequence of
20 shots, given a sequence of timed actions, is found below.

[0098] Let $A(i,j)$ be the cost for a transition from shot i to shot j . Let $B(i,t)$ be the cost for shot i at time t based on all actions occurring at time t . Let $C(\Delta t)$ be the cost associated with shot duration Δt . Let N be the number of time
25 intervals, M the number of available shot choices. Then the shot sequencing algorithm works by building an M by N table $BEST(i,t)$, containing the best partial cost leading to shot i ending at time t , with "back pointers" $ibest(i,t)$ and $tbest(i,t)$ pointing to the end of the previous shot. Thus,

the shot before (i,t) is $ibest(i,t)$ at time $tbest(i,t)$. The pseudo-code for the complete sequencing algorithm is as follows:

```

for all shots  $i$  and  $j$ 
5      create table of shot transition costs  $A(i,j)$ 
for all shots  $i$  and times  $t$ 
      create table of shot costs  $B(i,t)$ 
for all times  $t$ , for all shots  $i$ 
      initialize  $SUMB(0) = B(i,t)$ 
10     for  $\delta = 1$  to  $\delta_{max}$ 
           $SUMB(\delta) = SUMB(\delta) + B(i-\delta)$ 
          compute  $MIN(\delta) = \min(j) best(j,t-\delta) + B(j,i)$ 
          let  $JMIN(\delta)$  be the value that achieves the minimum
           $best(i,t) = \min(\delta) C(\delta) + SUMB(\delta) + MIN(\delta)$ 
15      $jbest(i,t) = JMIN(bestdelta)$ 
           $tbest(i,t) = bestdelta$ 

```

Finally, build the solution backwards by following back-pointers and generate EDL.

[0099] The best editing sequence can be found because a
20 search over all possible durations at each time step is performed. In another embodiment, a search for only the shot at the previous time step is performed.

[00100] Another embodiment of the method uses continuous takes from two or more real camera recordings of a live scene. In this case, the takes are synchronized, so that frames with the same time-code correspond to events taking place at the same time during recording. SDL descriptions of each take can be provided by way of frame-by-frame annotation. Many algorithms exist for tracking the locations, sizes and orientations of human faces in video. They provide the basis for creating the SDL descriptions using minimal user supervision (essentially, the labeling of actor's names and the correction of tracking errors). In yet another embodiment, the 3D motion of the real actors is recorded as well, using any of a variety of existing motion capture systems. The symbolic projection algorithm described above can be used to automatically generate SDL description of all shots, under the condition that the cameras must be calibrated to the motion capture system. This can be done using any of a variety of calibration or motion control systems. In both cases, shot descriptions in SDL can be used to compute the optimal editing and generate an EDL for automatically editing the recorded scene.

[00101] Given an animated scene, a script of actions, and an arbitrary set of cameras, the system computes an editing sequence which minimizes a global cost function. As explained above, the cost function can be written as the exponential of a sum of different functions multiplied by scaling coefficients which depend on application-specific cinematographic and editing preferences. Different choices of coefficients lead to different solutions. To resolve the issue, the system includes a method for choosing those

coefficients in a principled fashion, using annotated examples of fully edited movies typical of a given cinematographic and editing style. In one embodiment, the method requires that the example movies be broken down into
5 shots, using any of a variety of available cut detection algorithms. SDL annotation of the shots is made available, i.e. the screen positions, angles and sizes of all actors' faces in all shots must be given. Additionally, a script of actions is to be provided. From such input, the cost of each
10 example sequence can be computed as a function of the (unknown) scaling coefficients. A good choice of coefficient is one that maximizes the average score of all the given examples. Such good coefficients can be computed automatically, using any available machine learning method
15 developed for semi-Markov conditional random fields.

[00102] In one embodiment, the score of a shot sequence can be normalized by the sum of scores for all possible shot sequences compatible with the given action sequence and style parameters. The normalized values can then be interpreted as
20 conditional probabilities and the theory of semi-Markov conditional random fields can be used for learning the style parameters from the given examples. Generally, there is no closed-form solution for the optimal assignment of weights, so that the optimal assignment of weights is performed using
25 numerical optimization techniques. Examples of techniques that can be used are gradient descent algorithms and quasi-Newton methods, such as the L-BFGS algorithm. Other well-known machine learning techniques such as perceptron learning or entropy maximization are also applicable.

[00103] Parameters obtained in this manner can be packaged into a 'style-sheet' representative of the given examples. One embodiment includes a (separate) style-sheet for television talk-shows and classical Hollywood movies. Style-sheets for more specialized genres can also be created in the same fashion.

[00104] The method can be extended in several directions. For example, the method can use 'ideal' shot descriptions S_i in SDL that do not correspond to any existing camera, but can be used to suggest novel camera placements. It is then the task of the animator to create such a camera to generate the suggested edited sequence. The method can also be used to generate optimal sequences using any subset of available cameras. Users can also enforce constraints on the solution, for example by choosing a given camera shot S_u at time t_u for a given duration. The system then provides the best sequence that uses that shot, even if this solution is ranked lower than other possible solutions. Another feature of the system allows it to update its own internal parameters so that the solution proposed by the user receives a higher rank (user-profiling). Yet another feature of the system is to make it compute sub-optimal solutions in real-time on a frame-by-frame basis, and evaluate their scores. It is also possible to extend the system so that it handles gradual (rather than abrupt) transitions between cameras (dissolves instead of cuts) and between framings of the same camera (as with a moving or panning or zooming camera). In such cases, the duration of the transitions can be chosen according to the same methods that are used for choosing shot durations. The SDL language can also be extended to include lighting, color

and background information that can be computed directly from the image. Such additional information can then be used to evaluate the score of shot transitions. The proposed system uses a log-normal model of shot duration but other 5 statistical models can be used instead. Furthermore, it is possible to incorporate a model for short-term memory of previously viewed shots using a counter of the time spent in each shot. The editing model described herein evaluates the score of a transition between two shots but it can evaluate 10 all triplets of successive shots by computing the cost of a transition from shot S_i at t_i to Shot S_j at t_j to shot S_k at t_k . This feature can be used for trimming parts of the timeline, by evaluating transitions from one shot S_i at t_i to a 'garbage shot' S_j at t_j that trims all the frames for a given 15 duration, until the next shot S_k at t_k . This feature can be useful for removing slow, repetitive actions and for improving the quality of the transitions during fast actions.

[00105] The implementation described above is illustrated with examples taken mostly from two-speaker dialogs. The same 20 process can be used for scenes involving arbitrary numbers of actors and actions, such as but not limited to monologues, action scenes, fighting scenes with two actors, dialogs between freely-moving actors, scenes with three or more actors, etc.

25 [00106] It should be understood that the methods illustrated above may be executed by a machine provided with a processor and a memory. The processor is adapted to receive the inputs and configured to execute some or all of the steps of the methods described above.

[00107] Figure 3 illustrates one embodiment of a system 50 for editing an animation sequence. The system 50 comprises a memory 52, a shot scoring module 54, a transition scoring module 56, and an editing sequence generator 58. The system 50 receives as input animation events which are stored in the memory 52. Furthermore, the system 50 receives shot descriptions which correspond to the possible shots for generating the animation. The shot descriptions are also stored in the memory 52. The shot scoring module 54 and the transition scoring module 56 access the shot descriptions stored in the memory 52. The shot scoring module 54 is adapted to assign a score to each one of the shot descriptions in accordance with the corresponding animation event and shot rules. The transition scoring module 56 is adapted to assign a score to the transitions between shots throughout the sequence by analyzing the shot descriptions as a function of transition rules.

[00108] The shot descriptions, the transitions, and their associated score are then sent to the editing sequence generator 58. The editing sequence generator 58 is adapted to edit an optimal sequence using the shot scores and the transition scores.

[00109] In one embodiment, the system 50 is adapted to receive user's preferences about the editing style and the cinematographic style. These preferences are then used to generate shot and transition rules by the shot transition module 54 and the transition scoring module 56, respectively.

[00110] In another embodiment, the system 50 receives an exemplary edited sequence representative of a given editing

and cinematographic style which is stored in memory 52. The exemplary edited sequence comprises at least one action script, a set of exemplary shot descriptions, and an exemplary edit decision list. The shot scoring module 54 and the transition scoring module 56 are adapted to determine editing and cinematographic style parameters from the exemplary edited sequence and generate the shot rules and the transition rules, respectively, in accordance with the editing and cinematographic style parameters.

10 [00111] In one embodiment, the system 50 further comprises a shot description generator which is adapted to receive as inputs a set of positions and orientations for each one of a plurality of cameras involved in shooting an animation sequence, a layout of the scene for the animation sequence, including decor and elements therein, the trajectory of elements in the animation sequence, and the animation events. The shot description generator is also adapted to generate shot descriptions using the received set of positions and orientations for the cameras, the layout, the trajectory of the elements in the animation sequence, and the animation events.

[00112] In another embodiment, the shot description generator is adapted to extract the layout, the trajectory of the elements in the animation sequence, and the animation events from a script received by the system 50. The shot description generator creates the shot descriptions using the received positions and orientations for the cameras. Alternatively, the shot descriptions may be received by the system 50.

[00113] In one embodiment, the system 50 is adapted to receive user constraints such as a particular shot description to be inserted at a specific time within the sequence. The user constraints are received by the editing
5 sequence generator 58 which generates the optimal editing sequence taking into account the user constraints.

[00114] It should be understood that the above presented method and system can be used for "live" editing in which film time is equal to scene time. In "post-production"
10 editing, film time can be made faster or slower. While the present description refers to a process of editing a single scene based on a description of the actions and dialogs in the scene, assembling scenes together by use of flash-backs, cross-cutting between scenes, parallel editing, and the like
15 is also possible. In this case, a higher-level narrative strategy is given as an input, so that the cutting points between scenes can be determined.

[00115] The embodiments of the invention described above are intended to be exemplary only. Other examples, not
20 described in full details include a monologue on a stage filmed by M cameras; a sequence of player's actions from a game play which is edited into a sequence of 'highlights' from various camera angles; a 3D animated sequence generated by re-targeting motion from a motion capture session of a
25 real actor onto a synthetic cartoon character; a multi-camera recording of a stage performance, augmented with low-resolution motion capture so that at least the stage positions of all actors are recorded in synchrony with the film image. The scope of the invention is therefore intended
30 to be limited solely by the scope of the appended claims.

I/WE CLAIM:

1. A method for editing an animation sequence, the method comprising:

providing animation events;

providing shot descriptions corresponding to shots taken by a plurality of cameras;

assigning a score to each one of said shot descriptions in accordance with a corresponding one of said animation events and shot rules;

assigning a score to transitions between said shot descriptions throughout said sequence in accordance with transition rules; and

generating an optimal editing sequence using said score for said shot descriptions and said score for said transitions between said shot descriptions.

2. The method as claimed in claim 1, wherein said providing shot descriptions comprises:

providing a set of user-defined positions and orientations for each one of said plurality of cameras, camera parameters for each one of said plurality of cameras, a layout of a scene for said animation sequence comprising decor and elements therein, and a trajectory for said elements in said animation sequence; and

generating said shot descriptions using said set of user-defined positions and orientations, said camera parameters, said layout, said trajectory and said animation events.

3. The method as claimed in claim 2, wherein said providing a set of user-defined positions and orientations, a layout of a scene, and a trajectory comprises:

receiving said set of user-defined positions and orientations and said camera parameters from a user;

receiving a script of said animation; and

extracting said layout, said trajectory for said elements, and said animation events from said script.

4. The method as claimed in claim 1, wherein said providing shot descriptions comprises receiving a start time, a duration, an identification of one of said plurality of cameras, and a shot type, for each one of said shot descriptions.

5. The method as claimed in claim 1, wherein said providing animation events comprises receiving a starting time, a duration, an action and at least one role, for each one of said animation events.

6. The method as claimed in claim 1, further comprising receiving user preferences about editing and cinematographic style and defining said shot rules and said transition rules in accordance with said user preferences.

7. The method as claimed in claim 6, wherein said receiving user preferences comprises receiving at least one of action-specific preferences, transition preferences, preferences over a distribution of shot durations throughout said sequence, and scaling parameters controlling a relative weight of said user preferences.

8. The method as claimed in claim 1, further comprising:

receiving an exemplary edited sequence comprising at least one action script, a set of exemplary shot descriptions, and an exemplary edit decision list;

determining editing and cinematographic style parameters from said exemplary edited sequence; and

defining said shot rules and said transition rules in accordance with said editing and cinematographic style parameters.

9. The method as claimed in claim 1, further comprising providing a user with said optimal editing sequence and requesting validation from said user.

10. The method as claimed in claim 1, wherein said assigning a score to each one of said shot descriptions comprises assigning said score to each one of said shot descriptions as a function of at least one of a shot starting time, a shot duration, a shot type, and an association of said shot type with an action.

11. The method as claimed in claim 1, wherein said assigning a score to transitions comprises assigning said score to transitions as a function of at least one of a transition duration, and continuity and stylistic issues.

12. A method as claimed in claim 1, further comprising receiving user constraints and wherein said generating an optimal editing sequence is performed in accordance with said user constraints.

13. A system for editing an animation sequence, the system comprising:

a processor;

a memory accessible by said processor and adapted to store:

animation events; and

shot descriptions corresponding to shots taken by a plurality of cameras;

an application coupled to the processor, the application configured for:

assigning a score to each one of said shot descriptions according to a corresponding one of said animation events and shot rules;

assigning a score to transitions between said shot descriptions throughout said sequence in accordance with transition rules; and

generating an optimal editing sequence using said score for said shot descriptions and said score for said transitions between said shot descriptions.

14. The system as claimed in claim 13, wherein said processor is adapted to receive a set of inputs comprising user-defined positions and orientations for each one of said plurality of cameras, camera parameters for each one of said plurality of cameras, a layout of a scene for said animation sequence comprising decor and elements therein, a trajectory for said elements in said animation sequence, and said application is further configured to generate said shot descriptions using said set of inputs and said animation events.

15. The system as claimed in claim 14, wherein said processor is adapted to receive a script of said animation and said set of user-defined positions and orientations from a user and said application is further configured for extracting said layout, said trajectory for said elements, and said animation events from said script.

16. The system as claimed in claim 13, wherein said processor is adapted to receive a starting time, a duration, an action and at least one role, for each one of said animation events.

17. The system as claimed in claim 13, wherein said processor is adapted to receive a start time, a duration, an identification of one of said plurality of cameras, and a shot type, for each one of said shot descriptions.

18. The system as claimed in claim 13, wherein said processor is adapted to receive user preferences about editing and cinematographic style and said application is configured for defining said shot rules and said transition rules in accordance with said user preferences.

19. The system as claimed in claim 18, wherein said processor is adapted to receive at least one of action-specific preferences, transition preferences, preferences over a distribution of shot durations throughout said sequence, and scaling parameters controlling a relative weight of said user preferences.

20. The system as claimed in claim 13, wherein said processor is adapted to receive an exemplary edited sequence comprising

at least one action script, a set of exemplary shot descriptions, and an exemplary edit decision list, and said application is configured to determine editing and cinematographic style parameters from said exemplary edited sequence and generate said shot rules and said transition rules in accordance with said editing and cinematographic style parameters.

21. The method as claimed in claim 13, said application is further configured for providing a user with said optimal editing sequence and requesting validation from said user.

22. The system as claimed in claim 13, wherein said application is configured for assigning said score to each one of said shot descriptions as a function of at least one of a shot starting time, a shot duration, a shot type, and an association of said shot type with an action.

23. The system as claimed in claim 13, wherein said application is configured for assigning said score to said transitions as a function of at least one of a transition duration, and continuity and stylistic issues.

24. A system as claimed in claim 13, wherein said processor is adapted to receive user constraints and said application is configured to perform said generating an optimal editing sequence in accordance with said user constraints.

25. A system for editing an animation sequence, the system comprising:

a memory adapted to store:

animation events; and

shot descriptions corresponding to shots taken by a plurality of cameras;

a shot scoring module connected to said memory and adapted to assign a score to each one of said shot descriptions in accordance with a corresponding one of said animation events and shot rules;

a transition scoring module connected to said memory and adapted to assign a score to transitions between said shot descriptions throughout said sequence in accordance with transition rules; and

an editing sequence generator connected to said shot scoring module and said transition scoring module, said editing sequence generator being adapted to generate an optimal editing sequence using said score for said shot descriptions and said score for said transitions.

26. The system as claimed in claim 25, further comprising a shot description generator connected to said memory and adapted to receive a set of user-defined positions and orientations for each one of said plurality of cameras, camera parameters for each one of said plurality of cameras, a layout of a scene for said animation sequence comprising decor and elements therein, and a trajectory for said elements in said animation sequence and generate said shot descriptions in accordance with said set, said camera parameters, said layout, said trajectory, and said animation events.

27. The system as claimed in claim 26, wherein said shot description generator is adapted to receive a script of said

animation and to extract said layout, said trajectory for said elements, and said animation events from said script.

28. The system as claimed in claim 25, wherein each one of said shot description comprises a start time, a duration, an identification of one of said plurality of cameras, and a shot type.

29. The system as claimed in claim 25, wherein each one of said animation events comprises a starting time, a duration, an action and at least one role.

30. The system as claimed in claim 25, wherein said shot scoring module and said transition scoring module are adapted to receive user preferences about editing and cinematographic style and to generate said shot rules and said transition rules, respectively, in accordance with said user preferences.

31. The system as claimed in claim 30, wherein said shot scoring module and said transition scoring module are adapted to receive at least one of action-specific preferences, transition preferences, preferences over a distribution of shot durations throughout said sequence, and scaling parameters controlling a relative weight of said user preferences.

32. The system as claimed in claim 25, wherein said shot scoring module and said transition scoring module are adapted to receive an exemplary edited sequence comprising at least one action script, a set of exemplary shot descriptions, and

an exemplary edit decision list, and to determine editing and cinematographic style parameters from said exemplary edited sequence and generate said shot rules and said transition rules, respectively, in accordance with said editing and cinematographic style parameters.

33. The system as claimed in claim 25, wherein said editing sequence generator is connected to a user interface and adapted to provide a user with said optimal editing sequence and to request validation from said user.

34. The system as claimed in claim 25, wherein said shot scoring module is adapted to assign said score to each one of said shot descriptions as a function of at least one of a shot starting time, a shot duration, a shot type, and an association of said shot type with an action.

35. The system as claimed in claim 25, wherein said transition scoring module is adapted to assign said score to said transitions in accordance with at least one of a transition duration, and continuity and stylistic issues.

36. The system as claimed in claim 25, wherein said editing sequence generator is adapted to receive user constraints from a user interface and to generate said optimal editing sequence according to said user constraints.

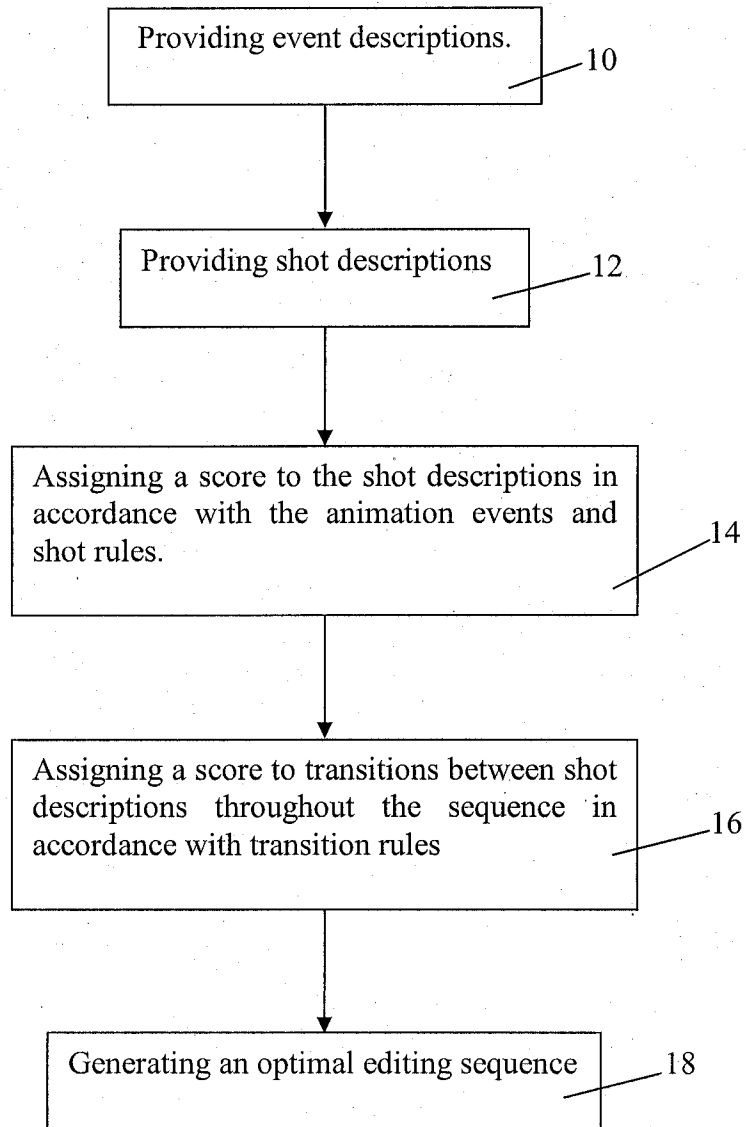


FIGURE 1

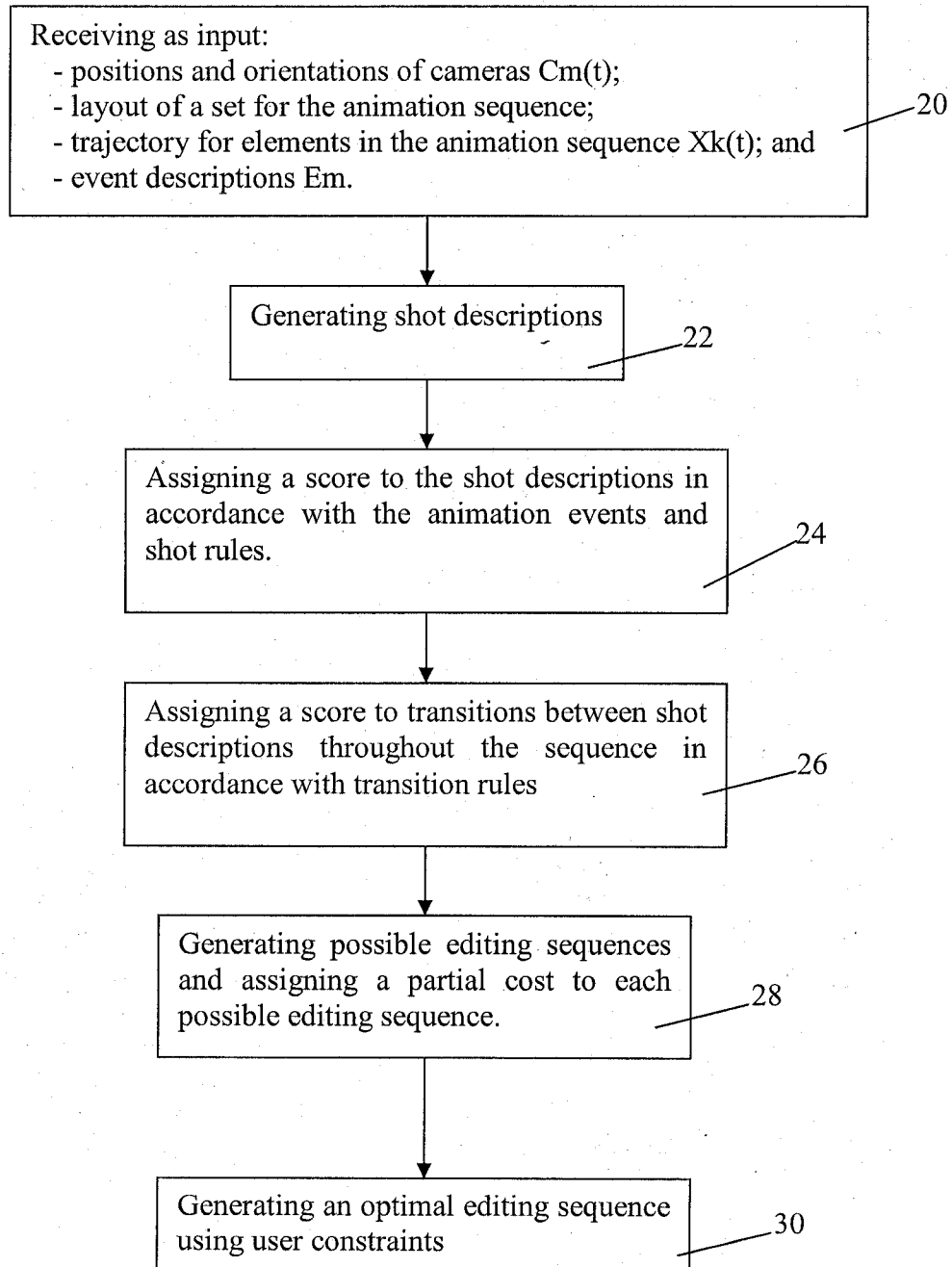


FIGURE 2

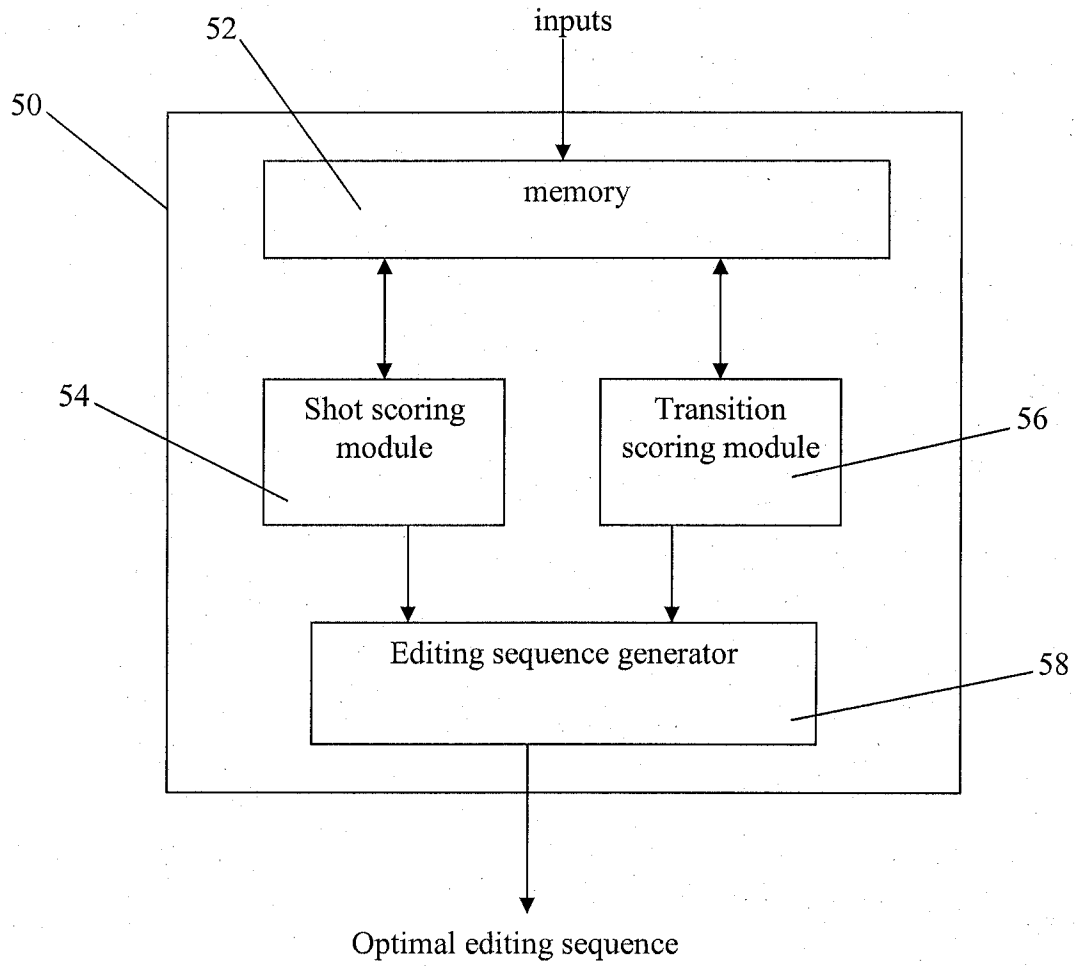


FIGURE 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2008/001925

<p>A. CLASSIFICATION OF SUBJECT MATTER IPC: H04N 5/265 (2006.01) , G03B 35/08 (2006.01) , G06T 15/70 (2006.01) , G11B 27/031 (2006.01) , H04N 13/02 (2006.01) According to International Patent Classification (IPC) or to both national classification and IPC</p>														
<p>B. FIELDS SEARCHED</p> <p>Minimum documentation searched (classification system followed by classification symbols) IPC: all</p> <p>Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched</p> <p>Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used) Canadian patent database, Delphion, Internet and WEST Some search term used: animation, edit, shot, score, transition, video, cinematography, optimal, sequence, rating, grade</p>														
<p>C. DOCUMENTS CONSIDERED TO BE RELEVANT</p> <table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th style="width:10%;">Category*</th> <th style="width:60%;">Citation of document, with indication, where appropriate, of the relevant passages</th> <th style="width:30%;">Relevant to claim No.</th> </tr> </thead> <tbody> <tr> <td align="center">A</td> <td>US 2007/0067724 (Takahashi et al.) 22 March 2007 (22-03-2007) see entire document</td> <td align="center">1-36</td> </tr> <tr> <td align="center">A</td> <td>WO 96/25710 (Mok et al.) 22 August 1996 (22-08-1996) see entire document</td> <td align="center">1-36</td> </tr> <tr> <td align="center">A</td> <td>US 6,807,362 (Girgensohn et al.) 19 October 2004 (19-10-2004) see entire document</td> <td align="center">1-36</td> </tr> </tbody> </table>			Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.	A	US 2007/0067724 (Takahashi et al.) 22 March 2007 (22-03-2007) see entire document	1-36	A	WO 96/25710 (Mok et al.) 22 August 1996 (22-08-1996) see entire document	1-36	A	US 6,807,362 (Girgensohn et al.) 19 October 2004 (19-10-2004) see entire document	1-36
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.												
A	US 2007/0067724 (Takahashi et al.) 22 March 2007 (22-03-2007) see entire document	1-36												
A	WO 96/25710 (Mok et al.) 22 August 1996 (22-08-1996) see entire document	1-36												
A	US 6,807,362 (Girgensohn et al.) 19 October 2004 (19-10-2004) see entire document	1-36												
<p><input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.</p>														
<table style="width:100%;"> <tr> <td style="width:50%;"> * Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed </td> <td style="width:50%;"> "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family </td> </tr> </table>			* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family										
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family													
Date of the actual completion of the international search 23 January 2009 (23-01-2009)		Date of mailing of the international search report 28 January 2009 (28-01-2009)												
Name and mailing address of the ISA/CA Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001-819-953-2476		Authorized officer C. Wong 819- 934-2666												

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CA2008/001925

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US 2007067724A1	22-03-2007	CN 1178469C CN 1332925A EP 1143714A1 WO 0040011A1	01-12-2004 23-01-2002 10-10-2001 06-07-2000
WO 9625710A1	22-08-1996	None	
US 6807362B1	19-10-2004	JP 2002077808A	15-03-2002