Narrative-Driven Camera Control for Cinematic Replay of Computer Games

Quentin Galvane
Rémi Ronfard
Marc Christie
Nicolas Szilas
Use of cinematic in video games is more and more popular
  - Necessity for Cinematic Replay of game sessions

A lot is made manually

How to produce a fully automatic cinematic replay?
➢ Control multiple dynamic cameras

➢ Create complementary, informative and well composed shots

➢ Edit them to produce a cinematic replay
**Related Work**

- **DCCL**
  - [Christianson et al. 1996]

- **Virtual cinematographer**
  - [He et al. 1996]

- **Darshak**
  - [Jhala 2006]

- **Behavior Trees**
  - [Markowitz et al. 2011]

**Idiom-based solutions**

- Require the definition of idioms
  - Setting the cameras
  - Selecting the appropriate moments

- Do not handle all the situations
  - No solution if the idiom doesn’t exist
REALTED WORK

Afterthought

Action-based solutions

Afterthought

Character-based solutions

Real-time Cinematography

Steering cameras

Film editing

Action-based solutions

Character-based solutions

Related work
Action-based solutions

Film editing

[Lino et al. 2010]

Character-based solutions

[Dominguez et al. 2011]

Action-based solutions

Character-based solutions

[Lino et al. 2011]

[Lino et al. 2013]
Key Idea

- Hitchcock principle:

  The size of a character on the screen should be proportional to its narrative importance in the story.

- Measure characters’ importance
- Use characters’ importance to
  - Control the cameras with steering behaviors
  - Perform the editing
Overview

Cinematic replay

Edit the camera rushes

Game

Record the Scenario and Animations

Cinematographer

Cam 1  Cam 2 ... Cam n

Specification

Rushes

Director

compute list of target characters

map to camera specifications

perform editing on rushes

Computed importances

Recorded Game

animation  scenario

Director

compute list of target characters

map to camera specifications

perform editing on rushes

Cinematic Replay

Game Engine

Game Recorder

Play time

Player
Cinematic replay

Edit the camera rushes

Camera shots
Animate the cameras

Cinematic replay
Edit the camera rushes

Narrative importance
Compute characters’ narrative importance

Target configuration
Extract targets

Camera Behaviors

Camera specifications
Retrieve the spec from the behaviors

Game
Record the Scenario and Animations

Director

computed list of target characters

Cinematographer

specifications

compute list of target characters

map to camera specifications

perform editing on rushes

Recorded Game
animation scenario

Play time

Player
Game Engine
Game Recorder

Replay time

Narrative importance
How to compute characters’ importance?

Classical two-level narrative structure:
- Beats that describe a narrative unit
  - [Mateas and Stern 2002]
- Atomic actions that compose a beat

Scenario
- Beats
  - Dialog
    - A speaks to B
    - B speaks to A
  - Fight
    - A moves to B
    - A kicks B
    - B punches A
    - ...
  - ...

Game
### Narrative Importance

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Game</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beats</td>
<td></td>
</tr>
<tr>
<td>Actions</td>
<td>Dialog</td>
</tr>
<tr>
<td>A speaks to B</td>
<td>B speaks to A</td>
</tr>
</tbody>
</table>

| A: Beat importance |  
|--------------------|---
| B: Beat Importance |   

| A: Action importance |   
|----------------------|---
| B: Action Importance |  

**Actions**

- A speaks to B
- B speaks to A
- A moves to B
- A kicks B
- B punches A
- ...
- ...
Computing Characters’ importance:

- Significance of a character in an action $S_c$
- Relevance of an action $R$

$$I_{beat}(c, t) = \sum_{a \in A_{beat,c}} S_c(a, t)$$

$$I_{atomic}(c, t) = \sum_{a \in A_{c,t}} R(a, t) \times S_c(a, t)$$
**Game**
Record the Scenario and Animations

**Narrative importance**
Compute characters’ narrative importance

**Target configuration**
Extract targets

**Camera Behaviors**
Retrieve the spec from the behaviors

**Camera specifications**
Animate the cameras

**Camera shots**
Edit the camera rushes

**Cinematic replay**
Analyze the cameras

**Director**
- Compute list of target characters
- Map to camera specifications
- Perform editing on rushes

**Recorded Game**
- Animation
- Scenario

**Player**

**Game Engine**

**Game Recorder**

**Replay time**

**Cinematographer**
cam1 cam2 ... cam n

**Play time**

**Cinematic Replay**
3 Types of targets

- PC: Player Character
- $P_1$: Primary non-player characters
- $S_1$: Secondary non-player characters
  - Defined using threshold on importance

At each frame

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$&lt; PC &gt;$</td>
<td>The player character is the only target</td>
</tr>
<tr>
<td>$&lt; P_0 &gt;$</td>
<td>One primary target that is not the player character</td>
</tr>
<tr>
<td>$&lt; PC, P_0 &gt;$</td>
<td>Two primary targets, one of which is the player character</td>
</tr>
<tr>
<td>$&lt; P_0, P_1 &gt;$</td>
<td>Two primary targets not including the player character</td>
</tr>
<tr>
<td>$&lt; P_0, S_0 &gt;$</td>
<td>One primary and one secondary target not including the player character</td>
</tr>
<tr>
<td>$&lt; P_+ &gt;$</td>
<td>One primary target or more</td>
</tr>
<tr>
<td>$&lt; S_+ &gt;$</td>
<td>One secondary target or more</td>
</tr>
</tbody>
</table>
**Camera Behaviors**

**Play time**
- **Player**
- **Game Engine**
- **Game Recorder**

**Replay time**
- **Recorded Game**
  - **animation**
  - **scenario**
- **Compute importances**
- **Director**
  - **compute list of target characters**
  - **map to camera specifications**
  - **perform editing on rushes**
- **Cinematographer**
  - **cam 1**
  - **cam 2**
  - **... cam n**
- **Rushes**
- **Cinematic Replay**

---

**Game**
- **Record the Scenario and Animations**

**Narrative importance**
- **Compute characters’ narrative importance**

**Target configuration**
- **Extract targets**

**Camera Behaviors**
- **Retrieve the spec from the behaviors**

**Camera specifications**
- **Animate the cameras**

**Camera shots**
- **Edit the camera rushes**

---
Definition of camera behaviors:

- Example of camera behavior: First person camera

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>MCU on PC 3/4 backright screenleft</td>
</tr>
<tr>
<td>&lt; PC, P₀ &gt;</td>
<td>CU on PC 3/4 backright screenleft and P₀ screencenter</td>
</tr>
<tr>
<td>&lt; PC, P₊ &gt;</td>
<td>CU on PC 3/4 backright screenleft and P₊ screencenter</td>
</tr>
</tbody>
</table>
Camera Behaviors

MS on Frank screen center and Lily 3/4 back left screen right in foreground

CU on Frank 3/4 back right screen left in foreground and Lili screen center

POV Lili CU on Frank screen center

POV Frank MCU on Lili screen center
**Camera shots**

- **Play time**
  - Player
  - Game Engine
  - Game Recorder

- **Replay time**
  - Recorded Game
    - animation
    - scenario
    - Compute importances
  - Director
    - compute list of target characters
    - map to camera specifications
    - perform editing on rushes

- **Cinematographer**
  - cam 1, cam 2, ..., cam n
  - Specification
  - Rushes

- **Cinematic Replay**

---

**Game**
- Record the Scenario and Animations

**Narrative importance**
- Compute characters’ narrative importance

**Target configuration**
- Extract targets

**Camera Behaviors**
- Retrieve the spec from the behaviors

**Camera specifications**
- Animate the cameras

**Camera shots**
- Edit the camera rushes
From PSL specification to camera placement

- Pruning Process:

For the `<size>` constraint

For the `<profile>` constraint

For the `<angle>` constraint

For the `<back|fore>` ground constraint
From PSL specification to camera placement

- Pruning Process:

MS on P1 screen left and P2 ¾ left screen right
Use of steering cameras to:
- Maintain shot specifications
- Perform smooth transitions between them

Two main forces implemented:
- Containing force
- Positioning force
Positioning force

- Ensure the framing and screen composition
- Composed of two distinct forces $F_1$ and $F_2$
  - $F_1$: Projection on the manifold surface
  - $F_2$: Steering towards the desired position
Camera shots - Demonstration
Cinematic replay

Game
- Record the Scenario and Animations

Narrative importance
- Compute characters’ narrative importance

Target configuration
- Extract targets

Camera Behaviors
- Retrieve the spec from the behaviors

Camera specifications
- Animate the cameras

Camera shots
- Animate the cameras

Cinematic replay
- Edit the camera rushes
Editing based on [Lino et al. 2011]

Implementation of cost functions to evaluate:

- Shots
- Cuts
- Pacing

Path finding in an editing graph
Results

- Nothing For Dinner

- 3 cameras behaviors:
  - First Person Camera
  - Second Person Camera
  - Master Shot

- 2 different styles:
  - Style #1: Point Of View Shots
  - Style #2: Over The Shoulder Shots
Results

Cameras:

Camera 1

Camera 2

Camera 3

Set of rules #1

Set of rules #2
Cinematic replay

Set of rules #1

Set of rules #2
Limitations

- Requires narrative information from the game
- No temporal ellipse or flashback

Future work

- More camera styles
- Validation and user studies
- Real-time in-game cinematography
Our narratively-driven system:

- Automatically generates cinematic replays of game sessions
- Works with large action vocabularies
- Smoothly animates cameras that cut well together
Thank you for your attention!

Narrative-Driven Camera Control for Cinematic Replay of Computer Games

Submission 19