Camera Control in Practice
Virtual cinematographer

• Placement of virtual camera
• You have access to scene geometry
  – But the scene is dynamic
• Versus first-person and third-person limited
4 basic shots with two actors

- Standard
- Over-the-shoulder
- Point-of-view
- Profile
How to show an actor

Extreme close-up
Close-up
Close shot
Medium shot
Full shot
Line of action

• Rule of thumb: never cross the line of action
• Imaginary line that keeps the audience oriented
Line of action example
Straight-forward camera control

class Scene
{
    List<Actor> actors;
    Matrix line_of_action;
}

class Actor
{
    Matrix orientation;
    Sphere body;
    Sphere head;
}

\[
\begin{bmatrix}
F_x & U_x & R_x & 0 \\
F_y & U_y & R_y & 0 \\
F_z & U_z & R_z & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]
Computing the line of action

1. \[ U = \frac{U_{\text{primary}} + U_{\text{secondary}}}{2} \]

2. \[ R = R_{\text{primary}} - R_{\text{secondary}} \]

3. \[ F = \frac{R \times U}{\|R \times U\|} \]

4. \[ U = F \times R \]

5. \[ M_{co} = \begin{bmatrix} F_x & U_x & R_x & 0 \\ F_y & U_y & R_y & 0 \\ F_z & U_z & R_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]
Computing the focus and distance

<table>
<thead>
<tr>
<th>Shot Size</th>
<th>Head/Body Interpolation ($k_{hb}$)</th>
<th>Screen Interpolation ($K_s$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extreme close-up</td>
<td>0</td>
<td>3/2</td>
</tr>
<tr>
<td>Close-up</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Close shot</td>
<td>$\frac{1}{4}$</td>
<td>1</td>
</tr>
<tr>
<td>Medium shot</td>
<td>$\frac{1}{2}$</td>
<td>1</td>
</tr>
<tr>
<td>Full shot</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

1. \[ C = k_{hb}C_b + \left(1 - k_{hb}\right)C_h \]
2. \[ r = k_{hb}r_b + \left(1 - k_{hb}\right)r_h \]
3. \[ d = \frac{r}{k_s h} \]

$h = \text{height of the frustrum view plane}$
Computing the angle (1 actor)

1. \[ V_{\text{offset}} = M_{CO} \times \begin{bmatrix} 0 & 0 & -d & 1 \end{bmatrix} \]

2. Rotate \( M_{CO} \) and \( V_{\text{offset}} \) around the up vector of \( M_{CO} \) by \( \alpha \)

3. Rotate \( M_{CO} \) and \( V_{\text{offset}} \) around the right vector of \( M_{CO} \) by \( \beta \)

<table>
<thead>
<tr>
<th>Shot Type</th>
<th>Angle (( \alpha ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>-45</td>
</tr>
<tr>
<td>Over-the-shoulder</td>
<td>75</td>
</tr>
<tr>
<td>Point-of-view</td>
<td>-90</td>
</tr>
<tr>
<td>Profile</td>
<td>0</td>
</tr>
</tbody>
</table>
Handling Obstructions

• Use ray tracing to check for interference
• Have a list of alternative shots ready to use
Intelligent camera control

• What are the short-comings of the previous method?
• Need more constraints
  – Analyze the constraints and search for a solution that meets all constraints
Camera constraints

• Every actor has a set of constraints
  – Range of vantage angles
  – Range of viewing distances
  – Preferred vantage angle
  – Preferred viewing distance
Viewing constraints in spherical coordinates

Actor in spherical coordinates

Occlusion in spherical coordinates

Merging actor and occlusion gives us a solution range
Searching for optimal viewpoint

• Now search the solution-consistent area for the \((\theta, \phi, d)\) closest to the preferred vantage and distance

• Convert back to Cartesian coordinates to place camera
References

• Straight-forward camera control:

• Intelligent camera control:
The virtual film crew

• Director
  – Makes decisions about the best way to shoot a scene.

• Cinematographer
  – Camera placement to carry out director’s orders.

• Editor
  – Decides how long scenes and shots should be to preserve a good flow of timing.
Virtual director

• In film, the director controls the scene and actors to achieve desired camera action.

• In games, the director has little control of action.
  – Determine which good camera shots are available
  – Think about all the ways to shoot an event
Virtual editor

• Decides which shots to use
• Unlike film, must do this in real-time
• Also choose transitions between shots
Sketch of a virtual director

- Events are constantly happening in the game
- Collect information about events
- Determine priority of events
- Provide editor with list of currently available shots

1. Relative shot priority
2. Timing information
   - Start
   - Estimated length
   - Decay rate
   - End condition
3. Shot information
   - Primary & secondary actors
   - Emphasis
   - Etc.
Shot timing

• Beginning time
• Decay rate
• End condition function
  – Cannot be terminated yet
  – Can be terminated if another shot is available
  – Must be terminated
• Which shot is running depends on dynamically changing priorities of all possible shots
Sketch of virtual editor

• Different editors have different preferences
  – $L_{\text{shot}}$
  – $L_{\text{scene}}$
• New shot is chosen when end condition of current shot changes
• Filter possible shots
  – Start time is not too far in the future
  – Current shot must terminate now and start time is now or recently passed
• Sort remaining shots based on quality
Sorting possible shots

\[ q(t) = p(t) \cdot c_{\text{scene}} \cdot c_{\text{actor}} \cdot P_{\text{length}} \]

\[ P_{\text{length}} = \begin{cases} 2L_{\text{estimated}} - L_{\text{shot}}, & \text{if } L_{\text{estimated}} > L_{\text{shot}} \\ L_{\text{shot}}, & \text{otherwise} \end{cases} \]

\[ c_{\text{actor}} = \begin{cases} 4, & \text{if both actors are the same} \\ 3, & \text{if one actor is the same at the same priority} \\ 2, & \text{if one actors is the same at different priority} \\ 1, & \text{if both actors are different} \end{cases} \]

\[ c_{\text{scene}} = \begin{cases} 1, & \text{if current scene } \neq \text{ new scene} \\ L_{\text{scene}} - t_{\text{scene}} + 1, & \text{otherwise} \end{cases} \]
Choosing transitions

• Between shots in different scenes:
  – Cut
  – Fade

• Between shots in the same scene:
  – If both actors are same or primary actor is same and line of action is similar: cut
  – If line of action changes significantly: move
Intelligent directors and editors

• Yes...
References

• Straight-forward director and editor:

• Intelligent directors and editors:
  – He, Cohen, & Salesin, “The virtual cinematographer” A paradigm for automatic real-time camera control and directing”, *SIGGRAPH* 1996.