

# Lights...Camera...Let's Have Some Action Already!

**E**XT. STREET — NIGHT — WIDE HIGH ANGLE  
Camera tracks BURKE as he walks down rain-drenched street. Light foot traffic. BURKE nods to a doorman and continues, turning right on Catalina Street.

CUT TO: TWO SHOT — BURKE AND STAND  
EMPTY SIDE STREET — MED. FULL OWNER

Camera tracks BURKE walking slowly up Catalina. Footsteps can be clearly heard following. BURKE notices and eyes dart but does not turn. Shot widens as he continues up street. ROCCO enters frame following BURKE casually strolling along smoking. BURKE turns to a stop in front of a restaurant, “examining” the menu. ROCCO comes to a stop in a doorway, puts out his cigarette with his shoe.

CUT TO: ULTRAVIOLENT — TERRIBLY FASCI-  
REVERSE — OTS TWO SHOT — MED. NATING STORY BEGINS...  
FULL

BURKE turns continuing down street. ROCCO starts to follow again. BURKE grimaces as footsteps resume. Camera tracks ahead of BURKE showing ROCCO over shoulder. Continues leading him until he comes to a newsstand. He stops and turns to pick up and examine a paper, footsteps slow behind him coming to a stop.

CUT TO:  
MED. CLOSE — BURKE

He picks up a magazine, eyes alert. Trying to catch a glimpse of his pursuer to his right out of the corner of his eye. Suddenly a hand grabs his left shoulder. He jumps.

CUT TO:

STAND OWNER  
I'm not running a library here,  
Mac. That'll cost you a nickel.

BURKE takes a beat, tucks the paper under his arm, and tosses the man a coin. Turns to continue down street. Footsteps continue behind him. He quickens his pace. The footsteps quicken. He breaks into a run. His pursuer starts to chase.

## From Big Screen to Game Screen

**A**ll right, so it's a cliché action sequence. I'm a programmer and this is not my latest screenplay. But that's not the point. A quality director can take these simple ideas and create tension, drama, and anticipation. These are exactly the qualities that pull people into a story. However, in interactive game applications, generating feelings of tension and drama is very difficult.

It's easy enough to create a cinematic cutscene that follows traditional filmmaking techniques. However, this yanks the player out of the interactive experience. Modern 3D game engines can create cinematic sequences within games, but most of the time these

sequences are completely scripted using traditional animation techniques. The sequence fires when the player enters a location, pulls a lever, or triggers some other mechanism. Once started, the sequence follows a deterministic path. The game designer now has a choice to make. The first option is to control the camera shots to show the drama, suspending the interactivity. Second, the player can maintain complete camera control and try to catch the action. This can create a great sense of “What's going on up there?” as you rush to find a viewpoint. HALF-LIFE used this technique very effectively. However, crucial information cannot be delivered in this manner as the player may miss it by spending too long studying the magnificent architecture.

The ideal solution would be to present the drama to the character as much as possible while allowing the player full control. It's clear to me that the camera system in a story-driven game needs to be a crucial character. It needs to be aware of what the player is doing, what is going on in the world. The camera needs to be “intelligent” enough to find the best viewpoint to show the player what's going on without ruining the dramatic element. To address this situation, I'm going to explore the idea of “camera AI.”

## Smart Cameras

**C**onventional game wisdom seems to hold that 3D real-time shooters must use a first-person camera while story-driven 3D action and adventure

*Jeff is the lead programmer and chief idea monkey at Darwin 3D, but what he really wants to do is direct. When not out pitching his latest spec script, he can be harassed at jeffl@darwin3d.com.*

games need to use a third-person camera. This may be the case. It is certainly true that the first-person point of view (POV) is not an effective movie storytelling paradigm. If you get the chance to see the film *Lady in the Lake* directed by and starring Robert Montgomery, certainly check it out. It is a very interesting moviemaking experiment. Montgomery filmed almost entirely from a subjective point of view. The only time you see the protagonist is in reflection. While compellingly different and not a bad movie in its own right, it shows dramatically why the first person is not the most effective method for conveying drama.

Take the rough scenario I outlined at the beginning of the column. Played in the first person, I would hear footsteps and need to swing the camera around to catch what was going on. All the subtlety would be gone. My shadow would either simply be hidden when I turned, or be caught diving behind a wall, cover blown. Likewise, for the newsstand sequence, I would need to rely completely on sound to convey the surprise from the stand owner grabbing my shoulder. While all these issues have solutions, the first-person POV certainly limits the possible options. With this in mind, let's take a look at automated methods for third-person POV cameras.

## The Shooting Gallery

To begin with, I need to create a frame of reference for all the shots I want to compose. Fortunately, many years of cinematography have provided a ready-to-use guideline for shot composition. Let me start with the framing of a shot for a single person. Obviously, the simplest step would be to divide the shots into long, medium, and close. A long shot would include the character and the

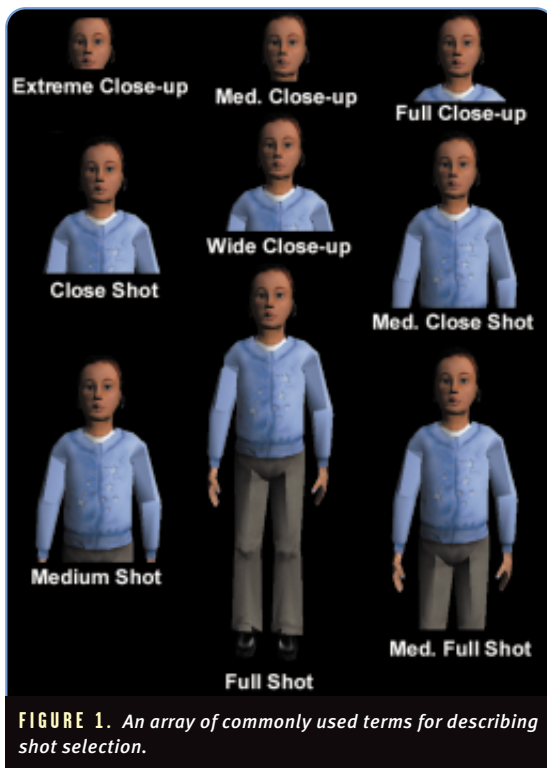


FIGURE 1. An array of commonly used terms for describing shot selection.

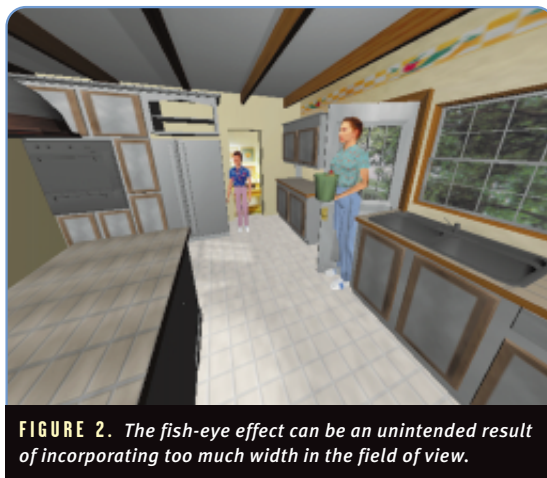


FIGURE 2. The fish-eye effect can be an unintended result of incorporating too much width in the field of view.

environment, a medium shot might be the character from the knees up, and a close-up would be just the head.

This doesn't really give the fine-grain descriptive terms that would be useful for framing a character. Typically, cinematographers frame the human figure using nine basic shots. The terms I am going to use are:

- Extreme close-up
- Medium close-up
- Full close-up
- Wide close-up
- Close shot
- Medium close shot
- Medium shot
- Medium full shot
- Full shot

You can see the framing for these shots in Figure 1. Once defined, these different shots are easy to work with in a real-time 3D game scenario. The goals are to center the character on the screen at the proper distance for the various shots. The first thing to find is the camera target, where I am going to point the camera. I could track a series of focus points on each character. Fortunately, I imbedded a skeletal system inside my characters so I can animate them. The base of each bone in my skeletal system is a ready-to-use focus point. I just pick the appropriate bone base to "look at" and use that as the camera target. For some of these shots, the focus point will be in between bones, but it's easy enough to interpolate the position between them.

Getting the right framing once I have the correct focus point is a bit trickier. I have two parameters I can play with. I can change the distance of the camera to the character or I can change the field of view (FOV) on the camera, effectively zooming in or out on the character. I have found that game players are very sensitive to FOV changes. If the field is too wide, the view takes on a fish-eye lens look (Figure 2) which can be very annoying (or cool, depending on your needs). If the field is too narrow, objects are hard to keep centered as subtle movements are exaggerated by the extreme zoom (think sniper rifle). So, I try to stay away from adjusting the FOV whenever possible. It's much better to move the camera. However, because we are in an interactive world, the

players can move around into situations where the camera is in a tight squeeze and pulling back is not possible without breaking through part of the set. Here, a little zoom now and then is a good thing.

So, using these simple guidelines, I started to create some “camera AI” routines. Given {Edit OK?} a character and a desired framing, the camera routine will start moving the camera to the desired position. This gives me the ability to say, “I’m ready for my close-up, Mr. Director Sir,” and it happens — all without any keyframes or scripting. But monologues are not very exciting.

### Let’s Strike Up a Dialogue

**A**dding a second character to the mix complicates things quickly. Let’s consider a scenario where a character walks up to another and starts talking. I start by tracking the main character with a full shot, as in Figure 3 for example.

As my character walks along, the other character can either approach me, or I can approach and start talking. At this point a dialogue is initiated and I need to start considering both actors in the scene as a pair. There is an imaginary line connecting the two. This line is actually very important as it defines a vertical plane that a camera cut between two cameras should not cross in most situations. I have seen several games and even some movies make this mistake. “Crossing the line” can really disorient the viewer because when a camera cut crosses the line, the relationship between the parties changes. The character on the right is suddenly on the left and your mind doesn’t immediately follow the motion path of the camera.

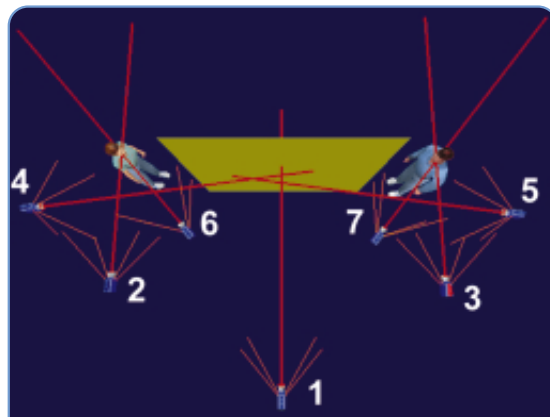
So, I start off with an establishing shot that shows the special relationship between the two characters. If I’m lucky, the standard single-

player camera shows the character adequately and can be used as an establishing shot. If for some reason the view of the second character is blocked, the camera needs to swing around to frame both characters in the view. I just spin the camera around the key character until both are in view. At this point, I can set up my dialogue cameras.

For two characters in a dialogue, I am going to set up a bunch of virtual cameras that I can cut between while the dialogue takes place. All of these cameras are on the same side of the “line.” I decide which side to stay on based on the initial positions, the direction each character is facing, and the environmental restrictions. Usually, there is a preferred side that is obvi-



**FIGURE 3.** Adding a second person requires careful shot selection to help viewers understand what is going on.



**FIGURE 4.** A camera setup for a two-person dialogue, offering a variety of camera cuts for different moods and effects.

ous given the conditions. The cameras I set up, shown in Figure 4, are:

1. Group profile: camera perpendicular to the line, framing both subjects.
- 2-3. Individual profiles: one camera framing each character.
- 4-5. OTS: over-the-shoulder shots of each subject.
- 6-7. Reaction shot for each subject.

Now, that looks like a lot of cameras. However, one thing I always find amazing about 3D games is that cameras are so underutilized. From a technical perspective, cameras are dirt cheap. A position, orientation, and field of view is all you really need. There is absolutely no reason for games to use the same camera, panning, swiveling, and gliding every-

where. Camera cuts are a very important part of storytelling. When is the last time you watched a film or television show that used a steady-cam following everyone around all the time? (Well, don’t count the opening sequence of *Touch of Evil*.) That is what we have currently in most games. (All right, mini-rant mode off.... Now back to our story.)

Using the action line between the characters and the character positions, these camera positions are calculated using simple 3D math. As a guideline, I have found that about ten degrees off the action line is good for the OTS cameras and 60 degrees is good for the reaction shots. The other cameras are just perpendicular to the line. You remember how to take the perpendicular to a vector, right? (Big hint: Swap X and Y and negate one.) During the scene, the action line can move if the participants move. Anytime that happens, the cameras just get recomputed.

Once all the cameras are set up, I need to determine which ones to use. This is where the camera AI comes in. The camera system needs to know about the characters. Who is talking? What’s the emotional state of each char-



acter? For example, when a character is talking, you probably want to use either the OTS camera or the profile camera that shows the speaker. While the character is speaking, you may occasionally want to switch to a reaction shot, particularly if the “mood” of the character dramatically changes based on the AI, script, or whatever. This is also where that real-time facial animation system you invested in earns its pay. Believable emotional reactions will really build the drama of the scene.

I used a very interesting reference source (see *The Virtual Cinematographer* in For Further Info) to set up a finite state machine that decides which camera to switch to depending on factors similar to the above, as well as delay timers. Another very good idea they suggested is to prompt the AI of the characters to move a little if they are too far apart or are blocking the camera.

4

## FOR FURTHER INFO

- Bares, William, Joël Grégoire, and James Lester, “Realtime Constraint-Based Cinematography for Complex Interactive 3D Worlds.” Proceedings of the Tenth National Conference on Innovative Applications of Artificial Intelligence, Madison Wis., July 1998. pp. 1101–1106. Available at <http://www.csc.ncsu.edu/eos/users/l/lester/www/imedia/papers.html>.
- He, L., M. Cohen, and D. Salesin. “The Virtual Cinematographer: A Paradigm for Automatic Real-Time Camera Control and Directing.” *Proceedings of Siggraph*. New York: ACM Siggraph, 1996. pp. 217–224.
- Katz, Steven D. *Film Directing: Shot by Shot*. Studio City, Calif.: Michael Wiese Productions, 1991.

### More Research on Cartoon Rendering

After the non-photorealistic rendering articles in February and March, I got a note from Adam Lake, one of the researchers at Intel who is working on NPR. He is presenting a paper at the Non-Photorealistic Animation and Rendering Symposium this summer covering more advanced silhouette and shading algorithms. He has graciously made it available to the public. You can get the paper at <http://www.cs.unc.edu/~lake>.

Though I haven’t tried that yet, it certainly makes a lot of sense.

## Two’s Company, Three’s a Crowd

Extending this idea to more than two subjects complicates the setup quite a bit. For three people, a line can be established between two of the participants based on who is speaking and who is facing one another. The general

patterns for a three-person conversation will fall into an “A” or “L” pattern depending on the layout of the participants and the action line chosen. In Figure 5, the people are in an “A” pattern leaving a nice action line between speakers A and C. However, there are also valid action lines between the other players. So what do you do? You can always transition to the establishing shot to reset the group. In general, however, when the action line changes, there will be a valid camera that was on the same side as the previous line, so that should be the first choice. You can further complicate these groups by adding individual reaction shots. However, the basic alignment should provide enough options.

For more than three speakers, it’s usually best to generalize the shots into group shots. You can create close-ups for the speaker and try to cut that with OTS shots from different sides of the group. Luckily for game developers, large groups of characters isn’t something we want a whole lot of anyway, for other reasons.

## Let the Player Have Creative Control

Everyone wants to be a director. This includes the game player. Most players will want to control the action. However, these techniques give the player a great deal of options beyond just spinning the camera around the key character, hoping to

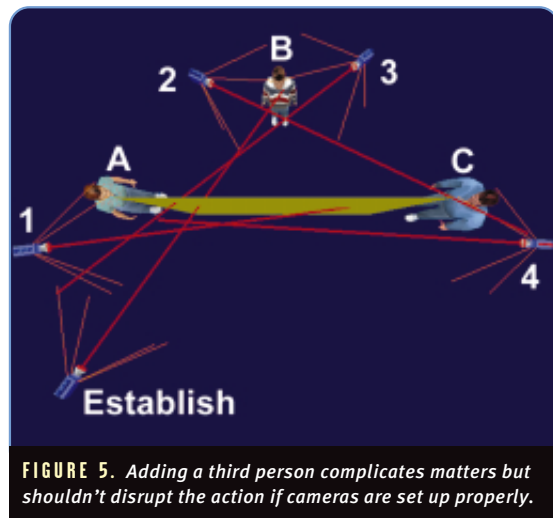


FIGURE 5. Adding a third person complicates matters but shouldn't disrupt the action if cameras are set up properly.

find a good view. You can allow the player to jump through your possible cameras, modify the view once the cut happens, or take total control preventing cuts. They could even assume first-person POV, if desired. The point is to provide options that allow you to tell a story without locking the player out.

Another interesting point is that the participants do not necessarily have to be people with whom the game player speaks. When the player goes to pick up an object, open a door, or look at a painting, the object in view can become one of the subjects in the dialogue. Looking over the shoulder of a character at a painting and then cutting back for the reaction shot would be very cool. Think of the AI reactions you could fire off.

As our games become more complex, it is definitely time to start thinking about more sophisticated camera usage. Applying some of the AI techniques we have been using for characters to cameras certainly makes a lot of sense. As a bonus, experimenting with these sorts of camera routines is pretty fun. It’s amazing how you can begin to influence mood and pacing by manipulating very few parameters. ■

## Acknowledgements

Thanks to Lisa Washburn at Vector Graphics (<http://www.vector.com>) and Steve Tice at QuantumWorks Corp. (<http://www.quantumworks.com>) for art contributions.