Cubism and Cameras: Free-form Optics for Computer Graphics

Andrew S. Glassner
glassner@microsoft.com

January 25, 2000

Technical Report
MSR-TR-2000-05

The camera model used to create synthetic 3D computer graphics is usually based on physical cameras. Cubist painters explored the process of creating images from multiple, simultaneous points of view. We believe that cubist principles can lead to a rich variety of interesting and dramatically useful idioms for illustration and storytelling, both in still images and in motion. We present the general idea, show some drawn images of the idioms, discuss the applications, and describe our implementation. We also discuss our plans for an improved interface for designers, directors, and other image-makers who do not wish to work directly with 3D modeling systems. We conclude with some preliminary results.

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

Introduction

Synthetic 3D computer graphics requires three basic elements to create an image: objects, lights, and a camera. Historically, the camera has received the least attention of the three.

Early computer graphics used the simplest camera model of all: the pinhole camera. A physical pinhole camera is nothing but a box. A flat piece of film is placed against one interior wall, and a pinhole is pricked in the opposite wall. When a piece of tape over the pinhole is removed, light from outside the box can pass through the pinhole and strike the film. The result is a picture with infinite depth of field, and a constant, uniform exposure of the film as long as the pinhole is not covered [4].

Several researchers have worked on more sophisticated camera models, including for example depth of field [9] and optical lenses [3] [8] [2]. Some of our prior work has looked at the effect of a shutter's shape and timing on the exposure [5].

But with rare exception [11] [6], computer graphics has assumed a camera model that was based either the single-point perspective of the pinhole camera, or on the orthographic perspective familiar to architects and technical drafters. In general, a single form of perspective is used for the entire image. Note that stereo images are usually generated by creating two independent, single-point perspective images, to simulate the two independent views seen by the two eyes.

These simple camera models are useful and practical tools, and allow creators to work with digital simulations of practical devices with which they are already familiar. However, computer graphics has the power to use imaging models that employ much more unusual optics.

We believe that such models can serve as much more than mere curiosities. Just as the cubist painters explored the forms of visual representation for painting, and found interesting new ways to communicate, so do we believe that free-form optical models can provide powerful new ways to communicate with synthetic images and animated sequences.

These techniques are not limited to synthetic images. The techniques of image-based rendering (IBR) can be used to take photographic source information and distort them to simulate the same non-linear camera optics.

The basic idea can be considered an animated, fluid form of cubism. The follow sections describe the idea in more detail, and give some preliminary results.

The Basic Idea

Creating images with nonlinear optics has been a powerful form of art for many years. In the fine arts, cubism served as a new and expressive way to represent the subjective interpretation of each viewer with respect to a common subject. It also allowed the artist to represent different parts of the subject, showing many simultaneous interpretations from different points of view.

In the practical arts, backgrounds for animated films have often appeared oddly distorted when viewed as a whole. But when a camera, looking at only a small piece of the image, moves across the surface, the result can make this simple pan appear like a much more sophisticated camera move [10]. Figure 1 shows an example of this sort of image.

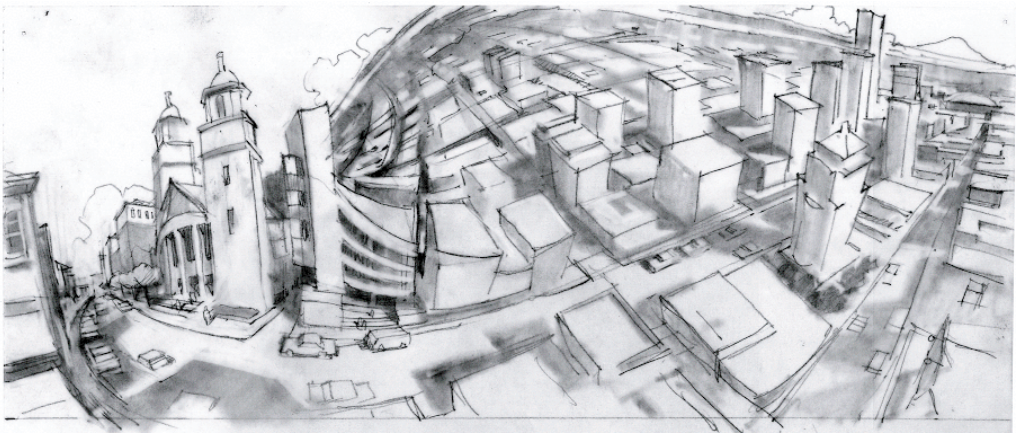

Figure 1: A non-linear street scene background for animation.

How might we create such a picture in a 3D computer graphics system? One approach might be to use a "slit-scan" technique, like that used for the feature film *2001: A Space Odyssey*. Although in practice this can be an elaborate process, the conceptual idea behind the slit-scan is straightforward.

Imagine a large piece of black cardboard and cut a narrow vertical slit that covers most of its height. The cardboard is then placed in front of a camera lens, so that the slit is to one side and the lens is completely blocked. Now slowly pull the cardboard from left to right as you move the camera. The result is that the film emulsion is exposed one vertical strip at a time. Since the camera is moving as the cardboard is moved, each vertical slice of the image could potentially be exposed from a camera location and orientation different than every other vertical slice.

If you spin the camera around its vertical axis while pulling the cardboard, you'll get a traditional panoramic image. More complicated motions can make the resulting image much more interesting, as though it was taken underwater, or through a shimmering haze of heat.

The slits-scan method makes a good starting point for describing our technique. To give things some specificity, imagine a hungry bear standing next to a river, waiting for a salmon to swim by. Eventually the fish arrives, and the bear lunges for it. We want to show the bear's lunge and capture of the fish. We could shoot this scene with traditional film techniques, using two cameras - one for an over-the-shoulder (OTS) shot of the bear, giving us the bear's point of view, and one camera in the water, giving us the salmon's point of view. When editing the footage, we could show just one or the other piece of film, or we could intercut between them, showing one and then the other in sequence.

It might be more dramatically interesting to show both at once. Let's imagine that we had a chain of a hundred cameras, each one placed right next to the other. The chain begins behind the bear's head, comes around the bear, plunges into the water, and turns around again to face the bear from underwater. If we fired off all hundred cameras at once, we could then compose a single image by taking a narrow vertical slice from each camera and pasting each slice side-by-side, as in Figure 2.

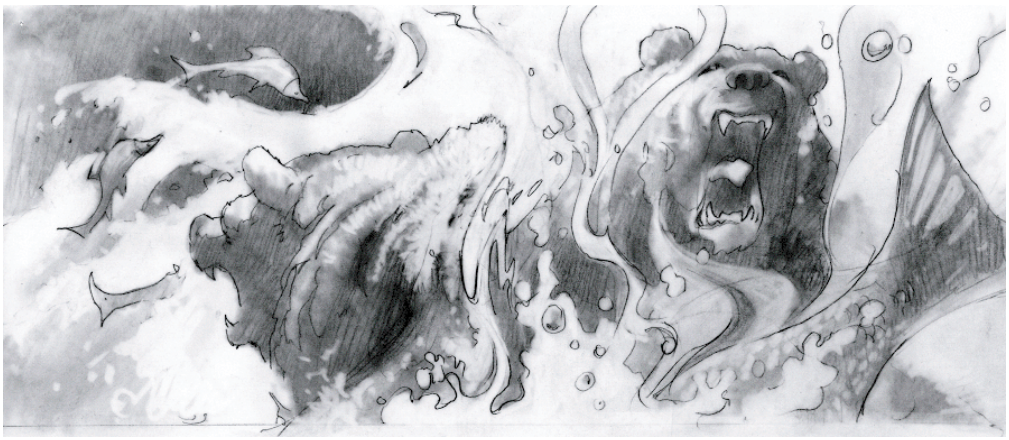

Figure 2: A bear going after a salmon. On the left, and OTS shot of the bear's view. The camera plunges into the water (note the bear's tail in the middle of the frame) and then turns up to see the bear in an OTS shot of the salmon's view.

This kind of technique has become popular in the last few years; a good example is in the feature film *The Matrix*.

If we imagine that each of these cameras is a motion-picture camera, then the image of Figure 2 is just a single frame from a piece of film or video. To my knowledge this hasn't been carried out in practice, probably because of the production costs (e.g. cameras, film, and processing) would be prohibitive.

This idea of a chain of cameras, whose images are sliced up and placed side-by-side, is at the heart of our approach. We generalize it below, ultimately to the point where every sample in the image can have a unique starting point and direction. We call our non-traditional imaging technique a *cubist cameras*.

Another way to show multiple points of view simultaneously is to use a "split-screen" technique, where for example one piece of film is played on the left and another on the right. This technique is often used to show both sides of a telephone call in film and television. In Figure 2, one could imagine that the boundary between the air and water is something like the artificial boundary in a slit-screen image, except that the interface is flexible and fluid, and the image is continuous.

We are interested in regions where the image is continuous. Whenever the creator of the image desires a visible seam or boundary, that can be easily handled by simply abutting the images where desired. The trick here is to make interesting changes in cubist camera position and geometry while maintaining a smooth and continuous image.

If we move from the physical world to the synthetic one, then we can place cubist cameras in places where it would be prohibitive or impossible in practice. Figure 3 shows the Seattle Monorail as it passes by a local landmark, the Space Needle. Once again, the image is continuous and simply a frame from a piece of video or film. However, there is again a distinct "seam" caused by the cubist camera passing through the window of the Monorail. This can be used for specific dramatic and storytelling effects.

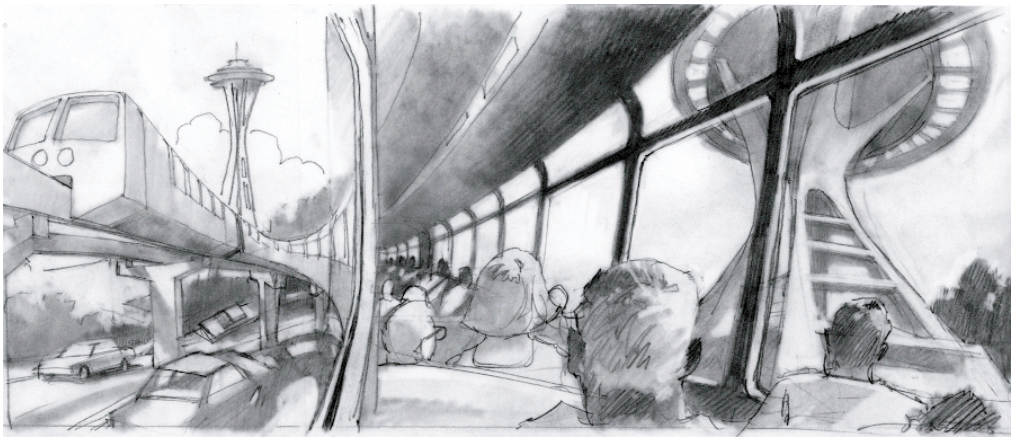

Figure 3: The Seattle Monorail. Note that the Space Needle is visible from both inside and outside the train.

For example, the interior and exterior of the Monorail can be slipped in time with respect to one another - by a second or by many hours. We can show how the day world and the night world exist simultaneously in space, but are separated in time, and do it with fluid, continuous imagery. It is this kind of expression that I believe makes cubist cameras so interesting.

The effects needn't be as big in scope as monorails, or as action-packed as lunging bears. Consider a phone call example. A young man and woman - let's call them Tony and Maria - are living across the street from each other. They wish they could call one another but are prohibited from doing so by their parents. They sit in their respective windows, quietly gazing at one another across the street. They are both awaiting the arrival of Tony's older brother, who has promised to speak to their parents and let the two lovers call each other. Tony's brother should be driving home any time now, but it's getting late and he's still not home. Tony and Maria watch each other and the road, each feeling a special bond with the other, quiet and sad, but also hopeful.

How might we show this scene in a film or television show? We could intercut between the two characters, or we could use a split-screen, or we could simply use one character's point of view throughout. Each of these approaches brings with it a particular kind of storytelling effect and statement.

Intercutting is an exciting, stirring effect, and is often used in suspense and action sequences to heighten the tension. It would be inappropriate for our quiet, contemplative scene. A split-screen effect could work, but it would intensify the separation between the two characters - we want the audience to feel the emotional bond that transcends the physical gulf that separates them. A single point of view could be intimate and melancholy, but would focus our attention on one character at the expense of the other.

Cubist cameras give us more storytelling choices. Figure 4 shows an example of how we could shoot this scene with a cubist camera. We see both the boy's and girl's point of view simultaneously, and we are also close to each one at the same

Figure 4: A boy and girl, lovers in spirit, but separated by parental authority and a wide, wide road.

time. We perceive the emotional size of the road that separates them, and we can keep an eye out for Tony's brother's car while staying close to the main characters. Note that we can see each character through the window of the other. As always, this should be interpreted as a single frame with continuous imagery from a longer piece of film or video.

As a final example of the storytelling possibilities created by cubist cameras, consider the following scene from a suspense movie focusing on industrial espionage.

Our heroine, Allison, has stolen the microfilm from her company's vault in Building A. She has arranged to make visual contact with Bruce as she walks the skybridge from Building A to Building B, and give him a sign; Bruce is waiting on the ground outside, looking up at the bridge. Allison walks out as planned, and turns to Bruce. She gives him the sign, but notices an odd look on his face. As she watches him, he starts to wave at her. She doesn't understand. What Allison doesn't know is that Bruce has seen the evil Carson casually walk out of Building A with a hypodermic needle in his hand, and he's arrogantly strolling across the bridge towards her. Bruce is trying to warn her, but she doesn't understand, and Carson is almost close enough to inject her.

Consider again the possible ways to shoot this scene. We have three characters, each with their own unique point of view. Each sees something important that the other two don't: Allison sees Bruce waving at her, Bruce sees Carson descending on Allison, and Carson sees Allison and the fact that there's nobody else in the tube. The scene is unfolding quickly. How can we show all of this simultaneous action?

Of course there are many choices, and many of those are clear and emotionally powerful. We believe that the cubist camera approach is a useful addition to the repertoire of good solutions to this problem. Figure 5 shows one way to do it.

Figure 6 shows an overhead, schematic view of how a physical chain of cameras might be placed in a real environment to capture this image. Note that the cameras would need to tilt up and down as well. We could create Figure 5 by abutting vertical slices from each of these cameras side-by-side. Of course, in the physical

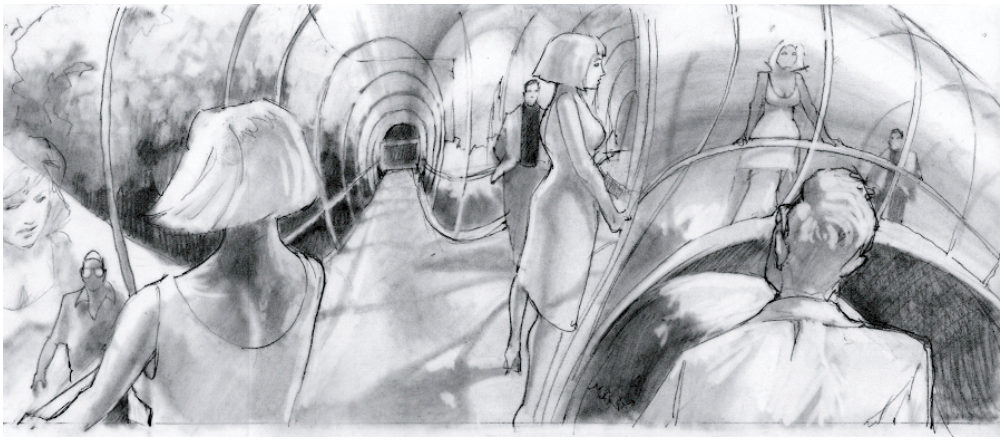

Figure 5: A suspense scene involving Allison in the tube, Bruce on the ground, and Carson approaching Allison with evil intent.

world there would be lots of practical issues to consider, such as alignment, mutual imaging of the cameras, registration errors, synchronization, etc. In the synthetic world, a single cubist camera does the trick without error.

Implementation

We have implemented cubist cameras as a material plug-in for 3D Studio Max. The material is based on the ray-tracing system for Max.

The cubist camera consists of two NURBS surfaces, which we call the *eye* and the *lens*. Each of these surfaces is a single continuous NURBS with U and V surface parameters that run from $(0, 0)$ to $(1, 1)$.

The process of rendering with these surfaces is very simple. A standard Max camera is created and placed somewhere off to the side of the scene. Directly in front of the camera, and covering its entire field of view, is a single rectangle.

When the system renders, it throws samples from the camera into the scene. Each one of these hits the rectangle, which has been assigned our cubist camera material. To shade the rectangle, our plug-in material is called.

The first thing we do is find the sample's (u, v) co-ordinates on the image plane (or the screen), and then offset and scale them so they run from $(0, 0)$ to $(1, 1)$. We then evaluate the *eye* and *lens* surfaces with those co-ordinates, creating two points. We trace a ray from the *eye* point to the *lens* point and into the scene. This ray is shaded in the normal way. The resulting color is passed back to the Max camera as the final, shaded color of the rectangle at that point. The color is placed into the image buffer and Max moves on to the next image-space sample.

Figure 7 shows the process. The renderer is filling in a pixel in the middle of the upper-right quadrant of the screen. Let's suppose this point has (u, v) co-ordinates $(0.75, 0.75)$ as measured from the lower-left corner. The yellow ball indicates the point on the *eye* NURBS that results when we evaluate that surface with those co-ordinates. The *lens* surface is in blue, and similarly the blue sphere represents

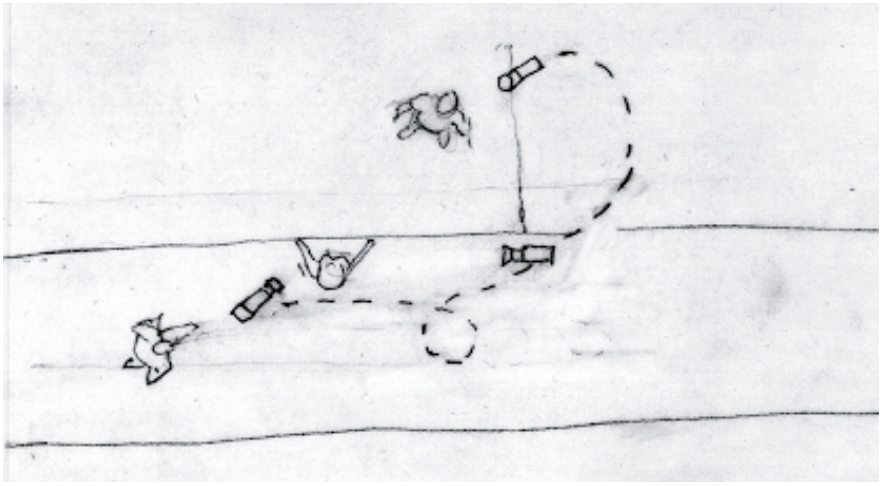

Figure 6: A schematic representation of the cubist camera for Figure 5.

the location of the co-ordinates $(0.75, 0.75)$ on that NURBS. The purple arrow represents the ray, starting at the yellow point on the *eye* surface and traveling in the direction of the blue ball on the *lens* surface.

Two points bear noting. First, the *eye* and *lens* surfaces are both invisible to the ray-tracer; that is, they are not rendered, they do not cast or receive shadows, and so on. Second, the point on the *lens* surface is used only to provide the ray's direction. If an object is sitting between the *eye* and *lens* NURBS, it will be intersected by the ray as usual.

Preliminary Results

To test these ideas, we have written a short film that takes place in Sheila's Diner. The production drawings for Sheila's are shown in Figure 8 and Figure 9. We have modeled the diner in 3D Studio Max; a rendered result is shown in Figure 10.

Our first example of a cubist camera will be represented by two flat sheets that have been curved around the counter. The eye sheet runs from the floor to nearly the ceiling, and parallels the outside of the L-shaped counter. The lens sheet is just inside the eye. The result is like a slit-scan camera that is swept along the counter, from the short leg of the L, around the corner, and along the long leg of the L, rotated so that it is always pointed towards the stools and the counter. Figure 11 shows the result.

Let's make a slightly more complicated version. This time we'll bend the top of each sheet over, so that the rays at the top of the image look down onto the countertop. This would be very difficult to achieve with a physical camera. Figure 12 shows the setup in wireframe mode; the black lines represent the NURBS surface of the *lens*, and the brown lines are the *eye*.

Figure 13 shows the rendered view. The vertical strips in the top half are due to shiny plates on the countertop.

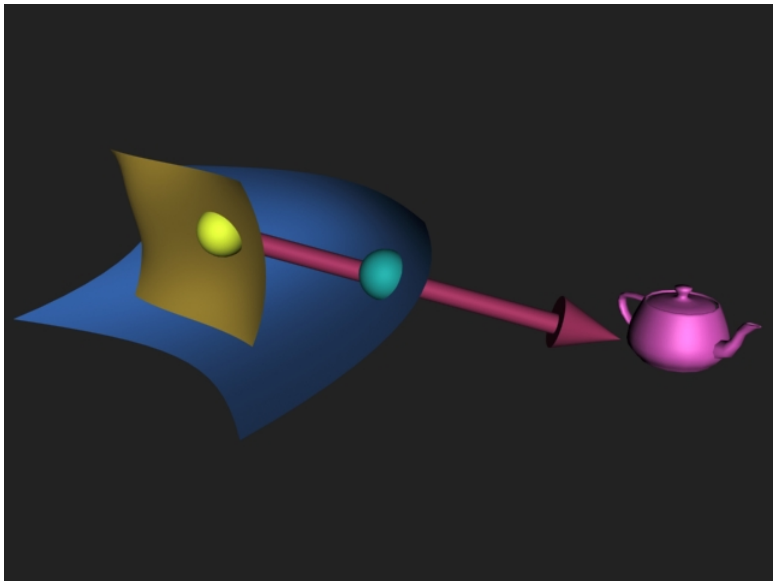

Figure 7: The *eye* surface is yellow. The *lens* surface is blue.

Figure 8: Sheila's Diner.

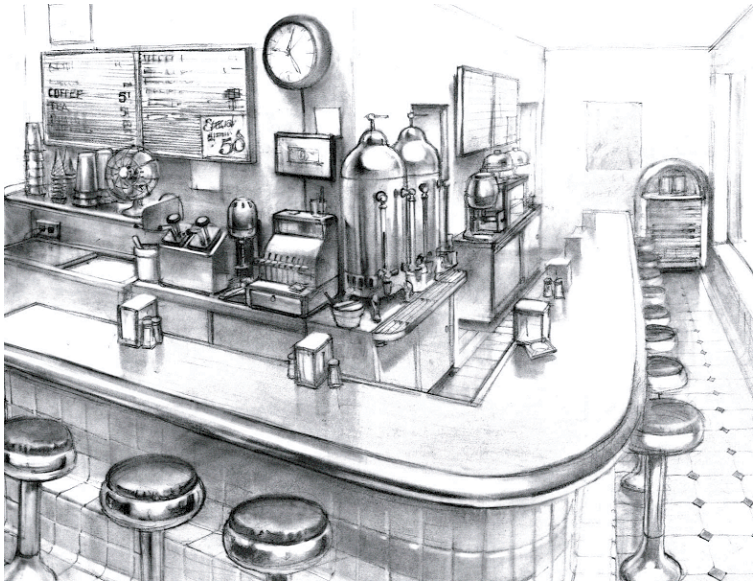

Figure 9: Sheila's Diner.

Figure 10: Sheila's Diner, modeled and rendered

Figure 11: Sheila's Diner from a simple cubist's point of view.

We are currently planning better camera positions for still images, and have already developed a script and storyboards for a short film that takes place in the diner, that can be well-told with cubist camera techniques. Figure 14 shows one example of the diner scene, with Studio Max standins for two diners and a waitress.

Other Applications

This technique may be used to accomplish some tasks that are currently handled by special software or algorithms.

To create real-time panoramic images, such as those used in Quicktime VR [1], one can simply create two concentric cylinders; the inner cylinder is the *eye*, and in the limit could have a radius of zero. The outer cylinder is the *lens*, and its radius is immaterial, as long as it is outside the *eye*. Then rays are fired in a radial pattern around the central column, creating a panoramic image.

If a designer wishes to use complex, realistic lenses [8], then the effect of those lenses can be precomputed and "burned-in" to the shape of the two cubist camera surfaces. Any desired geometric distortions, such as barrel or pinch, could be layered directly on top of the precomputed shapes.

To create environment maps [7] requires only two concentric cubes, like the concentric cylinders used for panoramas. The mapping from the image plane to the cube's surfaces would need to take into account the final arrangement of the six environment views desired in the final environment map.

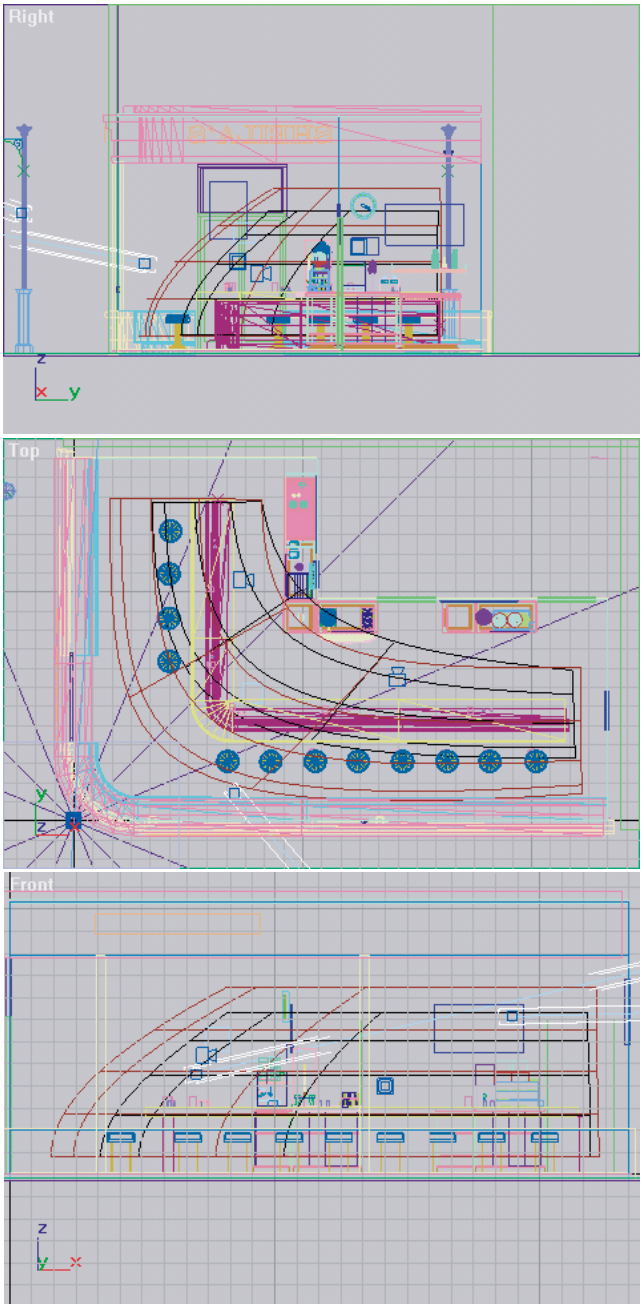

Figure 12: The curved cubist camera for Shiela's Diner is shown in black (for the *lens* and dark brown (for the *eye*). The views are from the right, top, and front arranged on the page from top to bottom.

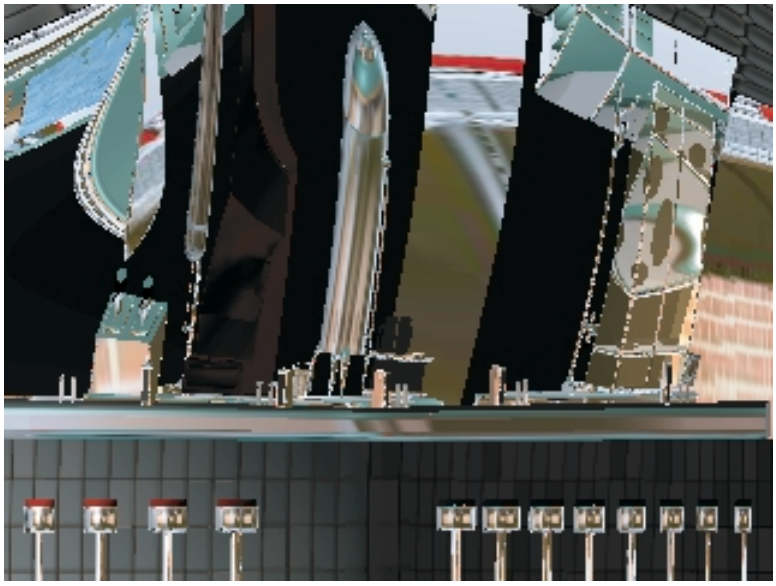

Figure 13: The view from the cubist camera of Figure 12. The strip-like reflections are caused by shiny plates on the counter.

Figure 14: Another cubist camera view of the diner, with standins for two patrons and a waitress.

Discussion

There is no particular need for the cubist cameras to be simple, as in the last section or the applications above. In fact, some of the most exciting applications for dramatic use involve using complex surfaces.

For example, suppose that we are in a courtroom scene, and the defendant is wringing his hands as testimony is given. We might know, as the audience, that this person nervously pulls on his wedding ring when someone gets too close to a secret he's guarding. The cubist camera might start out as a simple pair of parallel rectangles, shaped so that they represent a simple single-point perspective pinhole camera facing the character. But as he begins to work his ring, one corner of the surfaces begins to stretch and bend around, so that they are causing the rays to be cast from above his hand and down to the desk on which it is sitting. The view is thus "stretched" across the image; in the upper-left we have a frontal view of the defendant's nervous face, while in the bottom-right we have a little piece of image that appears to be taken from a point a few inches in front of his nose, and directed downwards. The transition zone is smooth and continuous, and the distortion from the simple camera to the complex one will also be smooth and continuous, and can take as long or short a time as desired.

One can raise the objection that such odd distortions could lose the viewer. That's certainly a possibility, and if the technique is pushed too far then that will probably happen. But if it is used with restraint, audiences will probably be able to understand what's going on, and they will gradually come to accept this device in the same way that they accept such odd devices as cuts and montages.

To push the technique, imagine a scene where a criminal is running away from the police. He's rushing down a stairway on the side of a building, past the windows on each landing. At one point he turns to look out the window, and we look over his shoulder at a window in the building across the way. The character is wondering if there's someone there he can take as a hostage. As we watch over the criminal's shoulder, the region of the frame occupied by the far window grows to take up more of the frame. It might not even distort at all - it just pushes the rest of the image out of the way. With the increased resolution of this frame, we can see with much more clarity. A secretary walks into the office on the far side of the distant window, and then starts to walk through a door into a hallway. Perhaps we have set up the character as someone who dreams about other people's lives, and imagines what they're doing when he can't see them. We might represent this dramatically by actually following the secretary through the door, down the corridor, and into another corridor, where she sits down at her desk. The outer region of the frame is still an over-the-shoulder shot of the criminal looking across the alleyway, and the central region is a view of the secretary. In between, the image might be very compressed.

Now someone important to the scene steps out of a door on that distant hallway; we could open up that region of the image to focus in on that person, and follow as they walk down the hall and into the original office, perhaps compressing the view of the secretary, or pulling the camera back from its long stretch down the corridor, like a balloon tightening up.

Taken too far, the cubist camera could be nothing but a confusing gimmick. But used with care, and making sure the audience is attuned to its use through gradual

introduction and development, cubist camerawork can become another useful device for effective visual storytelling.

Future Work

The biggest challenge in this approach is the user interface - modeling and designing NURBS surfaces is not a very convenient way to create a cubist camera, even for the technically inclined. For people without 3D modeling experience, it's probably useless to even try.

The simplest way to specify a cubist camera is probably to use images to identify the starting position and tracing direction for each ray. For simplicity's sake, suppose we had five 24-bit images, and wished to use only one sample per pixel. Then for a given pixel, we would look up the 24-bit X coordinate for the ray's starting location in image 1, and the Y and Z coordinates in images 2 and 3. Similarly, we need only two spherical angles to orient the ray, and we'd get their 24-bit values from images 4 and 5. We could clearly use more or fewer bits per ray, and more or fewer pixels in the image to accommodate more or fewer samples per pixel. With a reasonable local mapping, this technique could be used for non-uniform distributions of rays on the image plane. We call these ray buffers. Ray buffers could be used instead of explicit NURBS surfaces for all the applications and examples in this paper.

But making these ray buffers would still be difficult if they had to be hand-created. We instead propose a collage scheme. The idea is to render the desired scene with a few traditional cameras, and then use any traditional image-processing tool to cut out pieces of the resulting images and lay them together into a new image. When the designer cuts a piece of image, he or she implicitly also is cutting the five ray parameters from the ray buffers associated with each sample in that image, though they would probably be invisible in the user interface.

The designer then makes a collage of these image fragments. They could overlap, if desired. But probably much of the image would be blank. We propose using interpolation methods to smoothly fill in the missing data in the ray buffers, and to smooth the edges. Thus when the scene was re-rendered with the cubist camera, the patches that were pasted down by the designer would be imaged with that camera model in that region of the screen, and smoothly blend to other camera models at other locations. Whether the system actually makes the NURBS surfaces associated with the *eye* and *lens* surfaces is not important; they could be implicitly defined simply by the values in the ray buffers.

For animation, the collaged regions placed by the designer may be moved over time. Or, the designer may select a screen-space segment of an animated piece. Combining these, the designer may choose to draw a time-varying region over a piece of animation, and place that into the camera-control collage, which is itself a time-varying piece of animation. For each frame of rendering, the collage is constructed, the ray buffers are built, and the image is rendered.

The camera control and ray buffers may be used to directly control a synthetic 3D rendering as in our preliminary results. However, they may also be used to guide the reconstruction of a scene from photographic data, manipulating several streams of film or video using the techniques of image-based rendering. The result

can be pure film or video, purely synthetic, or any desired mixture of the two, such as photographed actors interacting with digital sets and objects.

Acknowledgements

Thanks to Tom McClure for the pencil illustrations, and for creating Sheila's Diner. Thanks to Dan Robbins for modeling, texturing, and lighting Sheila's in 3D Studio Max. Thanks also to Rick Szeliski for helpful discussions, and Steven Drucker for constructive criticism on this paper.

References

[1] Shenchang Eric Chen. Quicktime vr - an image-based approach to virtual environment navigation. *Proceedings of SIGGRAPH 95*, pages 29–38, August 1995.

[2] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics*, 5(1):51–72, January 1986.

[3] Jorge Alberto Diz, George Nelson Marques de Moraes, and Leo Pini Magalhaes. Simulation of photographics lenses and filters for realistic image synthesis. *COMPUGRAPHICS '91*, I:197–205, 1991.

[4] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics, Principles and Practice, Second Edition.* Addison-Wesley, 1990. Held in Reading, Massachusetts.

[5] Andrew Glassner. An open and shut case. *IEEE Computer Graphics and Applications*, 19(3):82–92, May 1999.

[6] Ned Greene. personal communication.

[7] Ned Greene. Environment mapping and other applications of world projections. *IEEE Computer Graphics & Applications*, 6(11):21–29, November 1986.

[8] Craig Kolb, Pat Hanrahan, and Don Mitchell. A realistic camera model for computer graphics. *Proceedings of SIGGRAPH 95*, pages 317–324, August 1995. ISBN 0-201-84776-0. Held in Los Angeles, California.

[9] M. Potmesil and I. Chakravarty. Synthetic image generation with a lens and aperture camera model. *ACM Transactions on Graphics*, 1(2):85–108, April 1982.

[10] Daniel N. Wood, Adam Finkelstein, John F. Hughes, Craig E. Thayer, and David H. Salesin. Multiperspective panoramas for cel animation. *Proceedings of SIGGRAPH 97*, pages 243–250, August 1997. ISBN 0-89791-896-7. Held in Los Angeles, California.

[11] G. Wyvill and C. McNaughton. Optical models. *Proceedings of CG International '90: Computer Graphics Around the World*, pages 83–93, 1990.