

# Andrew Glassner's Notebook

<http://www.glassner.com>

## Digital Cubism

Andrew  
Glassner

Where would computer graphics be without cameras? To make a synthetic picture with computer graphics, we usually imagine a camera of some sort, taking a picture of a scene. Such cameras range from the simplest pinhole camera to a sophisticated simulation of optics and shutters. Usually, though, our imaginary cameras are close analogs to the real thing.

If we're willing to move away from the idea of simulating a real camera, we can explore some interesting, alternative imaging models—such as digital Cubism. If we're thoughtful, we'll be able to harness possibilities for communicating ideas and story points in new and expressive ways. Let's look at some basic camera models, then consider how we can extend them.

### Pinhole cameras

You can see the traditional pinhole camera in Figure 1. It's probably the simplest form of a working camera, and it does indeed work. See the "Further Reading" sidebar for pointers to some Web sites of wonderful images made with pinhole cameras, and advice for building your own.

You can literally make a working pinhole camera out of almost any light-tight enclosure. A humble shoebox works well. In a shoebox camera, you typically mount the film on the inside bottom of the box, and poke a tiny hole in the top cover. You then cover the hole with an opaque shield, made of black paper or metal. To make an exposure, set the camera on the ground or a wall or some other steady surface, point the pinhole toward whatever you want to take a pic-

ture of, move the shield away from the pinhole so that light can reach the film, and wait a while. Then cover up the pinhole again. That's it for the exposure; now take your negative to a darkroom and develop it like any other photograph.

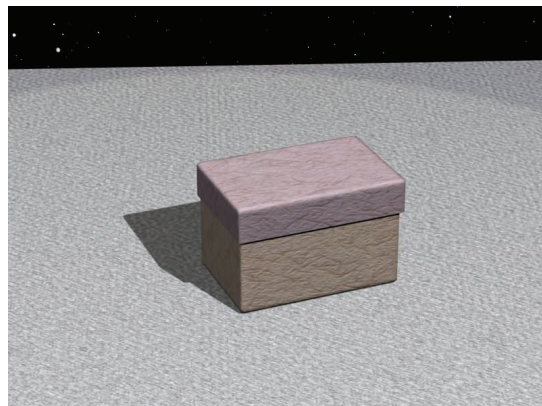
The optics of a pinhole camera are easy to visualize, as shown in Figure 2 (on p. 82). If we think of the pinhole as an ideal point, then for every point on the film there's exactly one ray of light that can pass from the scene, through the pinhole, and onto that point on the film. This observation is what made the early forms of ray tracing practical. Ray-tracing systems create *eye rays* (also called *screen rays*), which correspond to points on the film. Reasoning that any light striking a given point on the film must have arrived along the ray associated with it, the system follows that ray path into the environment, looking for objects and volumes that can send light back to the pixel along the ray.

The simple geometry of the pinhole model can get complicated when we start incorporating ideas like lenses and shutters. The basic idea remains the same, though: Rays of light in the scene strike the front of the camera and pass into some selecting and routing mechanism, which directs specific rays toward specific locations on the film.

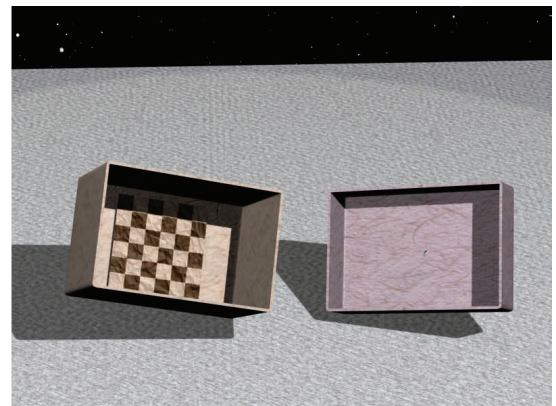
However, just as we've seen nonphotorealistic rendering methods in the last few years, we can also imagine nonmechanical camera models.

Perhaps the simplest variant on the shoebox pinhole camera is the sideways oatmeal-box pinhole camera, as Figure 3 shows. Here the film is no longer sitting

1 (a) A simple pinhole camera can be made out of any light-tight box. (b) The film is taped to the inside back wall of the box, and a tiny hole is poked in the front.



(a)



(b)

## Further Reading

Pinhole cameras are easy to make and they're fun to use. You can find information on building them from Kodak at <http://www.kodak.com/global/en/consumer/education/lessonPlans/pinholeCamera/>, or from the Exploratorium at [http://www.exploratorium.edu/light\\_walk/camera\\_todo.html](http://www.exploratorium.edu/light_walk/camera_todo.html).

Wonderful galleries of pinhole photography, along with advice and other resources, are available online at <http://www.pinhole.org>, <http://www.photo.net/pinhole/pinhole>, and <http://www.pinholeresource.com>. You can find instructions for building an oatmeal-box camera—along with lots of photos—at <http://users.rcn.com/stewoody/>.

A fascinating pinhole camera was printed in the Czech technical journal *ABC mlad\_ch techniku a prirodovedcu* (*An ABC of Young Technicians and Natural Scientists*) in 1979. Called the Dirkon, you only needed to cut the heavy pages out of a magazine and fold them up to make what looked like a 35-mm camera. But in reality it was a working pinhole camera, complete with a winding mechanism. You can download the plans for the Dirkon in PDF format from [http://www.pinhole.cz/en/pinholecameras/dirkon\\_01.html](http://www.pinhole.cz/en/pinholecameras/dirkon_01.html).

The geometry of a pinhole camera is pretty simple. You can see the details in my article "A Simple Viewing Geometry" (*Graphics Gems II*, J. Arvo, ed., Academic Press, 1991).

You can read up on the fascinating subject of maps and map projections, and see dozens of different projections, in J.A. Steers' *An Introduction to the Study of Map Projections* (Univ. of London Press, 1965). You can find a more compact description aimed at the computer graphics audience in Alan W. Paeth's "Digital Cartography for Computer Graphics" (*Graphics Gems*, Academic Press, 1990, pp. 307-320).

The panoramic camera has a long and colorful history, full of idiosyncratic inventors and rival patent wars. You can see a summary of the history of the subject at <http://www.panoramicphoto.com/timeline.htm>. I described Damoizeau's panoramic camera called the *cyclographe*; you can see a photo of it at [http://www.geh.org/fm/cromer-tech/htmlsrc/mD16300002\\_ful.html](http://www.geh.org/fm/cromer-tech/htmlsrc/mD16300002_ful.html). There's a brief, but clear, discussion of slit-scan techniques in <http://www.underview.com/2001/how/slitscan.html>.

Panoramic cameras are an important part of virtual panorama viewing environments, like the QuickTime VR system described in S.E. Chen's "QuickTime VR—An Image-Based Approach to Virtual Environment Navigation" (*Proc. Siggraph 95*, ACM Press, 1995, pp. 29-38).

In my discussion of Cubism, I referred to some well-known works. You can see Braque's *Houses at L'Estaque* online at <http://www.artchive.com/artchive/B/braque/housesle.jpg.html>, and Picasso's *Les Demoiselles d'Avignon* at [http://moma.org/collection/depts/paint\\_sculpt/](http://moma.org/collection/depts/paint_sculpt/)

[blowups/paint\\_sculpt\\_006.html](http://moma.org/collection/depts/paint_sculpt_006.html). An excellent book on Cubism is *Picasso and Braque: A Symposium* (Museum of Modern Arts, 1992), but any book on 20th-century art history will have something interesting to say on the subject.

If you want to think about putting real optics in front of your simulated cameras, there are lots of places to find guidance. Check out M. Potmesil and I. Chakravarty's "Synthetic Image Generation with a Lens and Aperture Camera Model" (*ACM Trans. on Graphics*, vol. 1, no. 2, Apr. 1982, pp. 85-108), J.A. Diz et al.'s "Simulation of Photographics Lenses and Filters for Realistic Image Synthesis" (*Compugraphics 91*, vol. I, 1991, pp. 197-205), and C. Kolb et al.'s "A Realistic Camera Model for Computer Graphics" (*Proc. Siggraph 95*, ACM Press, 1995, pp. 317-324).

Special-purpose real lenses have almost become their own field of study, particularly those used by Omnimax cameras and projectors. See N.L. Max's "Computer Graphics Distortion for Imax and Omnimax Projection" (*Proc. Nicograph*, 1983, pp. 137-159), as well as N. Greene's and P.S. Heckbert's "Creating Raster Omnimax Images from Multiple Perspective Views Using the Elliptical Weighted Average Filter," (*IEEE Computer Graphics and Applications*, vol. 6, no. 6, June 1986, pp. 21-27). Unusual projections are used now all the time for environment mapping; the seminal paper on that subject is N. Greene's "Environment Mapping and Other Applications of World Projections" (*IEEE Computer Graphics and Applications*, vol. 6, no. 11, Nov. 1986, pp. 21-29).

Perhaps the richest camera is the one that's defined by nothing more than some geometric equations or procedures. If you want to make an image with your film wrapped around a torus, for example, ray tracing is a great way to go. This is discussed and demonstrated in G. Wyvill's and C. McNaughton's "Optical Models" (*Proc. Computer Graphics Int'l 90: Computer Graphics around the World*, 1990, pp. 83-93).

Film grammar is evolving quickly. The basics, discussed in D. Arijon's *Grammar of the Film Language* (Silman-James Press, 1976), are expanding with every new technology. For some of the details on the bullet-time technique, see K.H. Martin's "Jacking into the Matrix" (*Cinefex*, Oct. 1999, pp. 66-89). The multiperspective camera technique is described in D.N. Wood et al.'s "Multiperspective Panoramas for Cel Animation" (*Proc. Siggraph 97*, ACM Press, 1997, pp. 243-250).

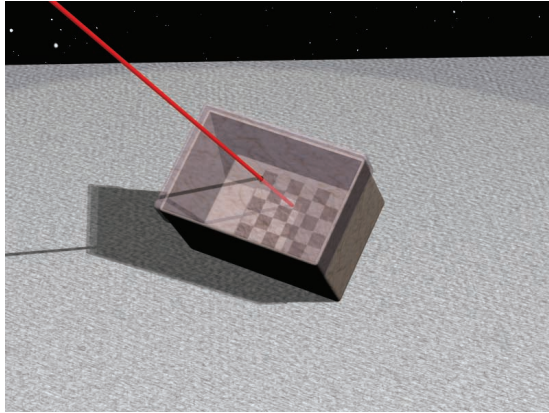
I've been thinking about the ideas I discussed in the main text for a long time. I wrote my first implementation of digital Cubism in 2000 when I worked at Microsoft Research. A summary of my work at that time is available in the technical report "Cubism and Cameras: Free-Form Optics for Computer Graphics" (Technical Report MSR-TR-2000-05, Microsoft Research, Jan. 2000; available at [http://research.microsoft.com/research/pubs/view.aspx?msr\\_tr\\_id=MSR-TR-2000-05](http://research.microsoft.com/research/pubs/view.aspx?msr_tr_id=MSR-TR-2000-05)).

against a flat wall, but is bent around the inside of a cylinder. Of course, the light entering the camera

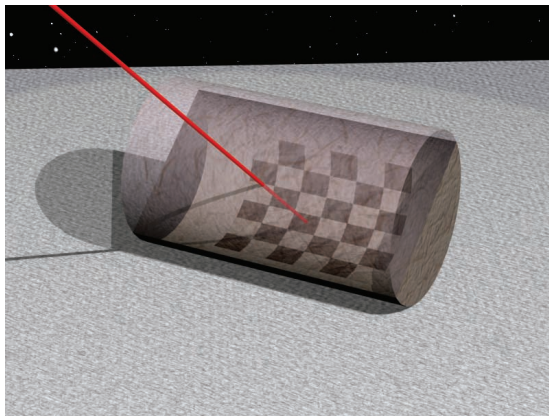
through the pinhole is unaffected by the shape of the film. Like before, rays of light pass from the scene,

through the pinhole, and strike the film. And as before, we can work out a specific one-to-one mapping from points on this cylindrical film to the rays that carry light to those points.

**2 Optics of the pinhole camera.** Each point on the film is illuminated by exactly one ray that passes from the environment and through the pinhole. I've made the lid transparent for this illustration.



**3 Oatmeal-box pinhole camera.** I've made the top half transparent for this illustration.

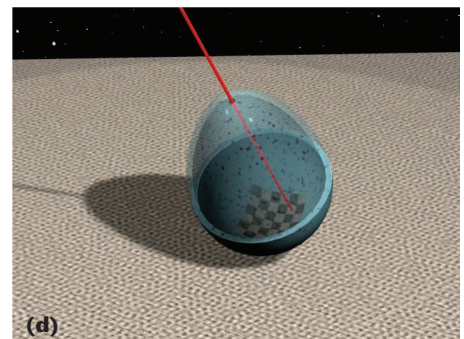
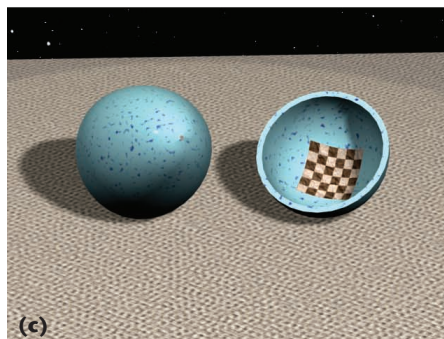
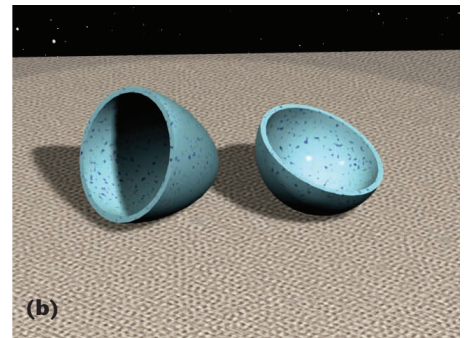
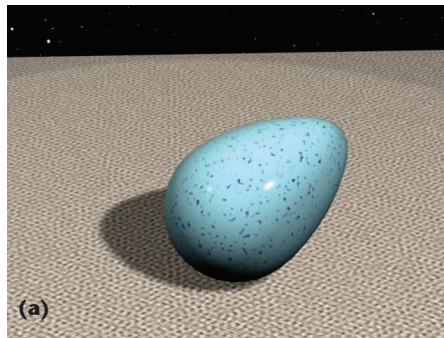


Bending a piece of film around the inside of a cylinder is pretty easy, but other shapes can be more difficult. The popular squishy goop known as Silly Putty is sold in a plastic egg (see Figure 4a) that separates down the middle (see Figure 4b). Although this egg is pretty small, there's no reason we couldn't drill a pinhole into the narrow end, and press a piece of film into the blobby end (see Figure 4c). Getting the film to fit smoothly would prove a bit tricky though, because a flat sheet would naturally bunch up and buckle as we tried to press it against the egg's inner wall. Rays pass through the pinhole to strike the film, as in Figure 4d.

This is where the computer graphics version of something is easier to manage than its real-world counterpart. Since our film is simulated, we can easily direct the light onto film that's been shaped any way we like, as long as we can write the equations (or procedures) that tell us which rays of light illuminate which spots on the film. If we can follow this up with a nice 2D mapping—or *unwrapping*—of the film onto a plane, then we can show the image as usual, although it will be distorted by the unwrapping.

Mapmakers carried out in-depth studies of the technique of unwrapping a curved surface onto a planar surface during the era of ocean exploration with tall ships. They needed to find ways to take the distributions of land, sea, wind, currents, and other phenomena on and near the Earth's surface and draw them onto sheets of paper, enabling efficient navigation of a ship at sea. The Earth isn't a sphere, but it's definitely round, and you can't represent it on a flat piece of paper without distorting it in some way. Cartographers came up with many possibilities for this transformation and each one had strengths and weaknesses, both geometrically and in terms of ease of use by naviga-

**4 (a)** Silly Putty egg. **(b)** Splitting the egg along its equator. **(c)** Making a pinhole camera out of the egg by poking a hole in the narrow end's tip and placing film inside the thick end. **(d)** Ray of light entering the egg pinhole camera. I've made the top part transparent for this illustration.



tors. The “Further Reading” sidebar offers pointers into this fascinating topic.

Cameras typically take a picture from a single point of view. However, what if we want to use multiple points of view in one image? The Cubist painters thought about that very question almost 100 years ago.

## Cubism

Cubism is the name given to a short-lived, but highly influential, style of art. Cubism peaked between about 1907 and 1914, mostly through the work of Georges Braque and Pablo Picasso.

Cubist art threw away traditional ideas of perspective, chiaroscuro, and other standard painterly techniques used in representational art, instead emphasizing the flat plane of the canvas. An aspect of Cubism that’s of value to us here is that Braque and Picasso often painted objects so that you could view them simultaneously from multiple viewpoints, overlapping or fragmenting different projections in the same painting.

The name Cubism derives from derogatory comments about Braque’s 1908 painting *Houses at L’Estaque* (see the “Further Reading” sidebar for pointers to online images of this and other paintings). Henri Matisse and Louis Vauxcelles dismissed the painting as nothing but a collection of cubes. It’s easy to see why they said this: Braque’s painting is made of simple geometric forms and flat coloring. Many historians point to Picasso’s *Les Femmes d’Alger (O. J. R. M.)* as the seminal work of what came to be called Analytical Cubism. In this painting, five female nudes have been reduced to angled, simply shaded forms. Analytical Cubism was characterized by the disassembly and simplification of forms, representing the subjects of the paintings and their environments with simple planes.

After about 1912, these painters produced works now referred to as Synthetic Cubism. The shapes were still simple, but they became rounder and less stark. Actual objects were pasted onto the canvas amid the painted objects.

By 1918 both painters had moved on to other styles, bringing an end to Cubism as a movement. The ideas of Cubism were influential though, and affected a wide range of other schools and styles of painting, including Futurism, Constructivism, and even Expressionism. Of course, I’ve barely scratched the surface of this subject; you can learn much more about Cubism and its related movements in almost any book on art history that includes the 20th century.

The aspect of Cubism most relevant to us here is its use of simultaneous, multiple points of view. Analytical Cubism asserted that no privileged point of view exists for an object, and that multiple views are appropriate—or perhaps even necessary—for understanding the structure of objects. That’s an interesting idea for looking at, say, a guitar or mandolin, but can it do anything for visual storytelling using computer graphics? You bet it can.

## Multicamera collages

Cameras are fascinating devices, and lots of people have used computer graphics to explore the possibili-

ties of algorithmically driven cameras that would be impractical or impossible in real life. Researchers and developers have studied and simulated complex, real-world camera designs, even including the special lens optics such as those used by Omnimax cameras and projectors. They’ve also studied entirely fanciful cameras like those that wrap the film around a torus (see the “Further Reading” sidebar for references to these projects, as well as others that I’ll discuss shortly).

By far the most common camera model used in computer graphics today is based on a flat sheet of film sitting behind a simple lens, controlled by a simple shutter. The shutter opens, light passes through the lens and onto the film, and then the shutter closes, completing the process. Of course, nothing is wrong with this model; it’s served us well for millions of still and animated images.

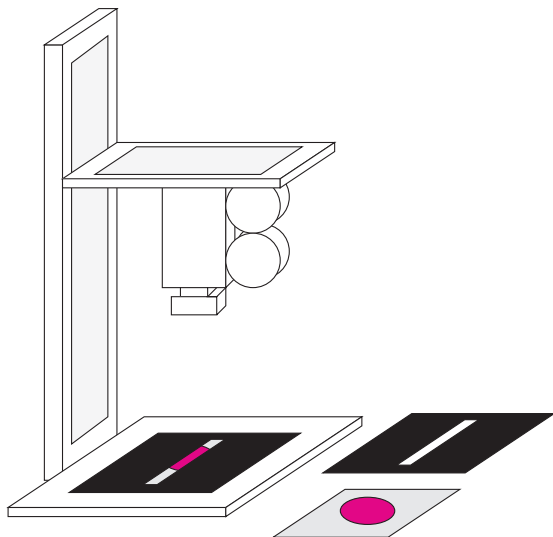
However, computer graphics can do things that no real camera could ever do. My original inspiration for exploring alternative imaging models came when I was thinking about how to shoot a scene in a film script I had written that required showing several events simultaneously. They were all in roughly the same location, but no single shot could have captured them all. I started to think about the different tools available to directors and cinematographers for addressing this problem.

Any time we talk about new visualization techniques for storytelling, it’s important to keep in mind that filmmaking is a craft of artifice. Today’s familiar grammar of film contains elements such as cuts, dissolves, and montages. The grammar of film is hardly static, though, and is constantly expanding as new technologies become available. Recent innovations include the bullet-time effect featured in *The Matrix*, the overlapping narrative structure of *Run Lola Run*, and the parallel imagery of *Timecode*. These are all mainstream films, but similar innovation is occurring in television, video, and animation.

What are our options when we want to show a scene with two or more simultaneous actions that are too far apart, or too detailed, to capture with a single camera? One way to show this action is to shoot the scene several times, with the camera in a different location for each take. Then the editor can cut the multiple pieces of footage so that the audience sees each one for a moment, then another, and another, and so on. Audiences are accustomed to the idea of cutting among multiple views of ongoing action, and it’s now a part of everyday filmmaking grammar.

Several other ways exist to show simultaneous action. A split-screen effect divides the screen into regions, each showing the same action from a different point of view. Typically black lines or boxes offset the regions from one another. Perhaps the first example of the split-screen effect used in an artistic way to help tell a story is Abel Gance’s classic film *Napoléon* (1927). More recently, it’s used from start to finish in *Timecode*. The split-screen is a powerful idea, but it’s typically used sparingly. One of its problems is that the black lines between the image regions can be distracting. Perhaps more troublesome is that the audience might find it difficult to interpret the relationships between the different scene views.

**5 Basic setup for slit-scan photography.** The camera is looking down on artwork that's sitting underneath an opaque sheet with a single slit cut into it.



Alternatively, the editor could choose to show the different takes of the same action consecutively. With this technique, we see one point of view and then another, but we're alerted through sound and visual cues that each shot is meant to occur at the same moment as the other shots. So the film presents multiple views of the same action, in effect rewinding and replaying the same segment of time over and over. We often see this kind of thing in horror movies, where some particularly gruesome event is shown from one person's point of view, and then again from another's—often in slow motion to make things that much more frightening.

Note that none of these techniques is realistic in the sense that it captures real life. Camera lenses do not see the world the way the human eye does, and even everyday cuts and wipes have no corollary to our normal visual experience. Audiences understand that film and video have languages of their own with their own rules, and as long as audiences are comfortable with those rules, they're willingly oblivious to the technique's artifice.

My goal was to develop something like a split-screen technique that didn't have those annoying black bars. But what could possibly replace them? One good candidate seemed to be those parts of the visual field that would naturally fit between the different pieces of the image, if we were somehow able to see them. Then we'd have a very flexible imaging tool: The image would be smooth and continuous, and the regions could change fluidly over time. I call this new technique the *multi-camera collage*, or MCC.

### Some film idioms

The multicamera collage is related to several existing techniques in filmmaking and computer graphics.

The closest work in filmmaking is the previously mentioned bullet- or stop-time effect. In this technique, a number of still cameras surround an environment, all looking inward at a central region. As the action takes place, all the cameras are fired simultaneously.

By sequencing the resulting images, the director can create a single shot that travels around the central region. The effect is that we're moving around a 3D tableau that appears to be frozen in time. We can use image registration and morphing techniques to smooth the transitions between frames. A straightforward but expensive variation of this technique uses film cameras rather than still cameras. The result is that we can freely manipulate time, running forward and backward at any speed, while the point of view moves around the scene.

Another way to capture spatially disjoint action in-camera (that is, without post-production effects) uses a technique called *slit-scan imaging*, made famous by Douglas Trumbull in the 1968 movie *2001: A Space Odyssey* (Trumbull was the special photographic effects supervisor for the movie). I show this idea schematically in Figure 5. In this arrangement, the camera is pointed down, shooting a piece of artwork held on an animation stand. In front of the artwork is an opaque piece of material that has a thin slit cut into it.

To shoot a frame of film, turn off all the lights in the room except for those illuminating the artwork. Open the camera's shutter and leave it open as you slowly move the camera itself, the artwork, and the slit from one position to another. When the motion is finished, close the shutter. You've now completed the exposure for that frame. Advance the film in the camera, reset the positions of the camera, the artwork, and the slit, and start the next frame.

Of course, all of these elements can be generalized: The slit doesn't have to be linear; we don't have to use a single slit; and the motion of the slits, art, and camera can be complex over the exposure's duration. We can change other imaging elements over time as well, for example by changing aperture or focus, or varying the illumination on the artwork.

This is a pretty complicated mechanical process: You could have three motors controlling the camera in as many directions; two motors controlling the motion of the animation stand; two more controlling the motion of the slit; and additional devices adjusting camera optics, lights, and so on. Everything has to be kept synchronized and calibrated over the course of a complete shot, which can involve hundreds of frames or more.

Computer graphics can simulate this entire process easily, and with absolutely precise repeatability. In one sense it's a little sad to lose the charm of a complex mechanical system that you probably could grow to love, but it's also great to know that with software, nobody can accidentally bump the project and ruin two weeks of work. Graphics also lets us easily add effects that would be difficult to accomplish in a mechanical system, such as the imaging of 3D objects (rather than an animation stand), motion blur, and even effects like lens flare.

The slit technique is also useful in other applications. Many panoramic imaging methods—such as those used for QuickTime VR—use a camera model that you can think of as a horizontally moving camera with a horizontally moving aperture made of a vertical slit.



(a)



(b)

**6** Two people having a conversation outside their neighborhood café. (a) View from the side. (b) Two-shot of the same scene.

We could also move the slit off of the animation stand and make it part of the camera. In 1889, J. Damoizeau patented a camera called the *cyclographe*, which he demonstrated in 1891 at a meeting of the Société Française de Photographie in Paris. The camera was specifically designed to take 360-degree panoramic photographs onto wide pieces of film that measured  $8.5 \times 80$  centimeters. The camera contained a built-in motor that you wound with a key, and a thin vertical slit just in front of the film. Once the camera was wound up, you placed it on a tripod and released the motor. The motor spun the camera in one direction while it wound the film the other way, past the slit. We might say that the camera imaged one vertical slice of the negative at a time as it turned through a complete circle. Damoizeau later built a second version of the camera that used a pair of mechanisms to shoot stereoscopic images (see the “Further Reading” sidebar for a pointer to a photograph of this remarkable camera).

Cameras used to digitize real objects—such as faces and 3D works of art—often turn this idea inside out. Rather than rotate in place to collect the environment that surrounds them, they orbit the object under study to get an image of it from all sides.

One of the more unusual imaging models in computer graphics is the *multiperspective panorama*, which recreates a technique that’s used in traditional animated films. To create what looks like 3D motion of the camera, artists paint a large background image that smoothly blends among numerous points of view. By panning a camera across the surface of this painting, the change in viewpoint can make it look like the camera is actually moving through space. It’s a difficult trick to pull off, but skilled background painters have learned how to do it beautifully. The multiperspective panorama system, or MP, was designed to create those kinds of backgrounds from full 3D models.

At a glance, the results of the MP system resemble the results of the MCC, which I’ll detail in the next section. But they’re different tools that produce quite different results. The most obvious difference is the audience isn’t supposed to see MP images all at once. Indeed, it’s the very fact that the camera moves over the image that creates the magic. On the other hand,

MCC images exist to explicitly address the desire to show multiple perspectives at the same time. Another important distinction is that the MP technique simulates the motion of a single camera over a single path, while MCC blends together multiple cameras operating independently.

Despite these differences, the techniques can be compatible. In fact, some MCC ideas nicely augment the MP, and vice versa.

### Putty cameras

Let’s imagine the most general geometry for finding the light falling on a piece of film. It would probably be an algorithm that took as input the  $(x, y)$  film location we want to fill in, and produced six real numbers: the three coordinates of a ray’s starting point  $A$ , and the three coordinates of its direction vector  $\mathbf{d}$ . All the cameras we’ve discussed so far can be written in terms of such an algorithm. In fact, many can be written as a straightforward mathematical formula.

But writing out algorithms and mathematical functions is hardly a good match to the types of mental states we’re in when working on a storytelling project. Maybe things would be easier if we could find a physical analogy for our camera optics, rather than abstract mathematics and computer science.

I’ll describe my first approach to addressing this problem. It didn’t work out in the long run, but it’s a great conceptual tool and helps us visualize the kinds of cameras that we could find useful.

Earlier I mentioned the plastic egg that Silly Putty comes in. Let’s open up the egg and think now about the putty itself. Suppose that instead of using a piece of photographic film for our camera, the film was made out of putty. Then we could stretch and bend the film into any shape we liked. Now let’s imagine that we could do the same thing with the lens. Then we could take our camera and stretch and twist it into any shape we want.

For example, suppose we want to film a conversation between two people facing each other over a table outside their local café, as in Figure 6. Normally we would take a 2-shot (as in Figure 6) of the couple, some over-the-shoulder shots of each person, and later edit these all together into a conversation. But suppose that for our storytelling purposes, we wanted to show both people from the front at that same time. We could of course

7 Both characters in Figure 6 are photographed from the front, and the two images are combined using a split-screen technique.



8 Inserting a sheet of putty film into Figure 6. The checkerboard texture is just to make the film sheet easier to see.



9 Rendered version of Figure 8 using the "porcupine's quills" model for the direction of light rays leaving the film.



create a split-screen, as in Figure 7, but that has all the problems of split screens that I discussed earlier

A putty camera would give us another option. Let's put a sheet of film putty into the scene as in Figure 8, and imagine that it's collecting light from rays that strike it perpendicular to its surface. In other words, think of the sheet of film as having a porcupine's quills sticking straight out of it at every point; those spikes are the direction of the light that strikes the film.

The image produced by this porcupine-putty camera is in Figure 9. Even this simple camera has the features that I'm after: the black line is gone, both characters are facing forward, and the space between them is imaged continuously and smoothly.



10 Adding a sheet of lens putty to Figure 6. The lens is the yellow and red checkerboard.

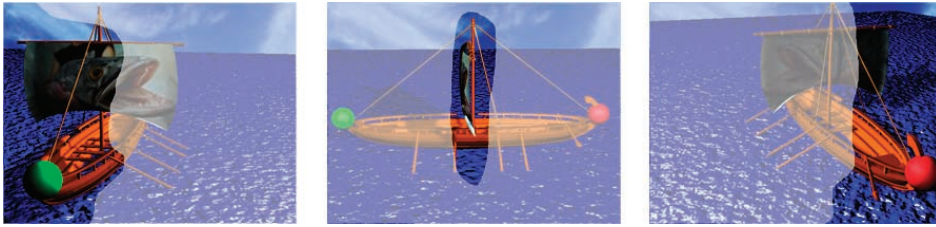


11 Rendered version of Figure 10. Compare this to Figure 9. Here we can see the front of the café. The windows and door are compressed, but visible.

Of course, we might want to have better control over the rays than just accepting whatever the porcupine's quills give us. Let's make another image with two flexible sheets instead of one. The first, called the film, will be the one we already have. But we'll add another sheet, which I'll call the lens. These are both flexible sheets with a mathematical description that supports a 2D parameterization (such as Bezier, NURBS, or subdivision surfaces). As long as I can give the sheet a pair of  $(u, v)$  coordinates and get back a 3D point on the surface, that's all I need.

Now to fill in a point on the film, I take its  $(u, v)$  coordinates to get the ray origin  $A$ , and use the same  $(u, v)$  values to find a point  $B$  on the lens surface. The ray heads off in the direction vector  $\mathbf{d} = B - A$ . Figure 10 shows these two surfaces in our scene, and Figure 11 shows a rendered image.

This technique is a good start, but it's got a few problems. First, positioning these sheets in 3D space is difficult, even using modern production systems. Second, picturing in our mind's eye how the sheets interact to determine the ray directions is just about impossible. Moving around control points on the two surfaces to get the rays to point where you want them is, in my experience, a never-ending task of endless tweaking. Every change throws off all the oth-



**12** Three images of a simple scene of a Greek ship at sea. Note the green ball at the front of the ship, and the red one at the back.

ers, and you can go around and around forever. It's hopeless to try making small or local changes.

The essential problem here is that we're trying to manipulate these 3D sheets to achieve a 2D result. Getting the surfaces to behave the way we want is difficult to visualize and control.

I don't think this two-sheet approach has much of a future as a user interface. It's a great way to think about the possibilities offered by this kind of camera model, but it's a lousy mechanism for interactively specifying it. Let's look at something better.

### The old collage try

Since our goal is to produce a 2D image, let's see if we can find a way of working that better matches the 2D results we're after.

The general approach that I'll describe here uses conventional cameras to create pieces of the final picture, which you then use to build a collage. The computer fills in the empty spaces in the collage.

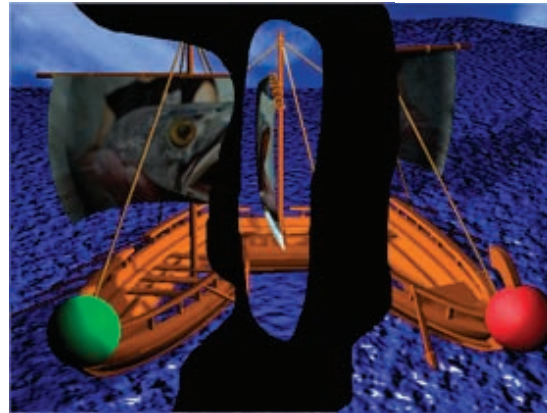
The general process goes like this (it's the same whether you're making a single still image or an animated sequence). Start off by rendering your scene from as many different points of view as you like, using your system's built-in cameras. Figure 12 shows the result for three different cameras. During this process, you tell the program to use a custom lens shader (which I call the *lens writer*). The shader has no visible effect on your rendering, but creates a couple of files per frame. Follow this up by running the camera tagger program. It takes no arguments and has no user interface; just run it, and when it's finished move on to the next step.

Now load the images you created in the last section into any image-editing program (such as Photoshop) that supports layers. Use a selection tool to cut away the parts of each image that you don't want in the final collage. Figure 13 shows such a collage using the images from Figure 12. Save the collage picture. Run another program, called the *lens builder*, and wait for it to finish.

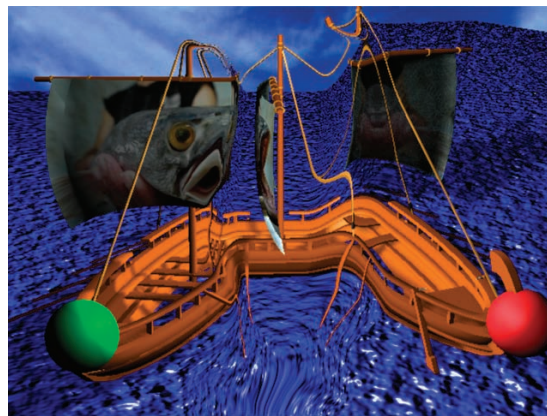
Now open up your 3D scene again, tell the renderer to include a different lens shader (which I call the *lens reader*), and render the image. This shader will create a new image that looks like your collage, but the black regions will be smoothly filled in with views of the scene, as in Figure 14.

If you're making an animated sequence, you'll probably want to make several collage files at different points in the sequence. The system will smoothly interpolate those collages along with everything else, producing a smooth animation with your changing cameras.

Note that there's no restriction on the types of cameras you use, how many you can use, or the shapes you



**13** Collage assembled from the three images of Figure 12. I overlapped the three images, and then drew a selection on each one by hand. I deleted everything outside of the selection for each image, leaving just the black background showing through.



**14** Re-rendered version of the collage in Figure 13.

pick from each one. You can pick one big blob from each camera, or a dozen tiny spots.

This technique has at least three nice things going for it. First, we don't have to accomplish anything difficult in 3D; we just use the regular 3D system with its built-in cameras, and make the images we want. Second, building the collage is an easy process in any 2D image-editing package that supports layers. Third—and maybe most importantly—the person using the system doesn't need to know anything about the underlying technical details. In the course of putting together a newly rendered frame, the user runs a couple of programs that manipulate files, but there are no user interfaces on those programs, and no variables to be selected or controlled. The programs do their work silently and independently.





15 Another example of the MCC. Three images of our couple chatting outside their café. Notice there's a mysterious pill floating in her drink.



16 Collage made from Figure 15.

17 Re-rendered version of Figure 16.



18 (a) A scientist working alone in her lab late one night. (b) The same scene, rendered with MCC.

Figure 15 shows three different views of a couple chatting outside a café, and Figure 16 shows a collage assembled from these images. Figure 17 shows the resulting re-rendered image.

Next time, I'll present the technical details of implementing the MCC, and how we can use it to create digital Cubism. To give you a taste of what we'll cover in the next installment, Figure 18 shows an example of the technique in action in a more complex shot. ■

**Acknowledgments**

Thanks to Steven Drucker, Ned Greene, Tom McClure, and Claude Robillard for discussions, ideas, and help.

Readers may contact Andrew Glassner at [andrew@glassner.com](mailto:andrew@glassner.com).