THE NEWSLETTER FOR LIGHTWAVE 3D ANIMATORS

LIGHTWAVEPRO

FROM THE PUBLISHERS OF VIDEO TOASTER USER MAGAZINE

Volume 1 Issue Number 9 • July 1994 • $8.00

Real World Effects

Inside:

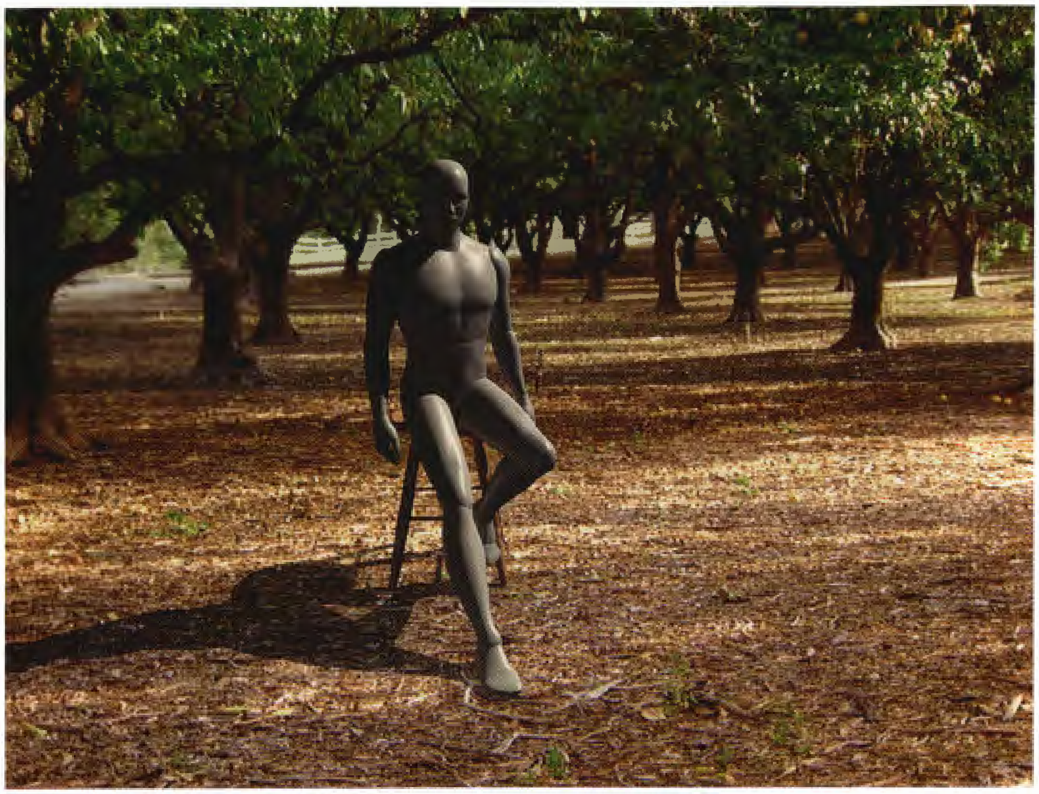
Let it Rain!

Realistic Beams of Light

Explosions

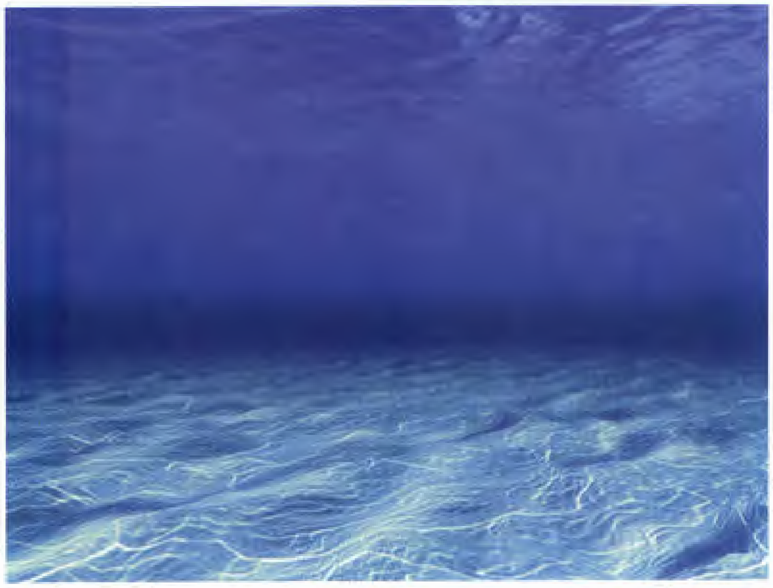
What's on the LIGHTWAVEPRO Disk?

See Page 17



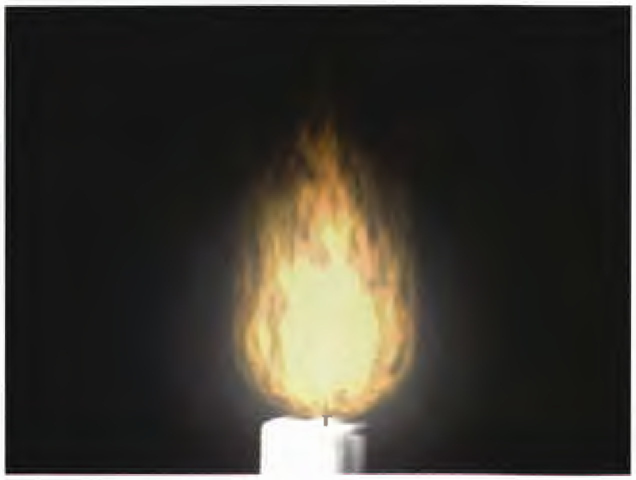
Sunshine

A captured video image (right) combined with the proper light intensities and colors help to create a realistic looking sunlit scene.
Copyright John F.K. Parenteau 1994



Water Surface

An image sequence applied as a bump map to the semi-transparent water surface allows a ray-traced light to simulate refracted water on the terrain below.
Copyright John F.K. Parenteau 1994



Candle Bonfire

While the flame appears more like a bonfire (right), flames can be changed dramatically by changing texture sizes and velocities.
Copyright Grant Boucher 1994

EDITOR'S MESSAGE

by John Gross

Computer animation has always had that 'computer' look to it. Recently though, advances in technology have given us incredibly realistic looking computer generated images, as evidenced by the mastery of *Jurassic Park* .

Just before 2.0 LightWave shipped, I remember the 2.0 Kiki image created by LightWave progrmmer Allen Hastings, which showed a glass on top of the older 1.0 Kiki picture. At first I thought it was a captured video image taken by Allen. Since it came from Allen, however, I knew it had to be LightWave generated.

Needless to say, I was quite impressed. I think that was one of the early 'photo-realistic' images that I had seen from LightWave. Sure, I had seen incredible spaceships and similar objects, from Ron Thornton, but I just knew that these were computer generated (where do you get footage of a spaceship?). The Kiki glass showed me what LightWave was capable of.

Until that time, most of my LightWave work was comprised of flying logos that screamed CGI! But that image taught me a lesson.

I can just imagine Allen getting a glass from a cupboard and modeling it. I'm sure he measured every possible measurement a glass could have. And I know that after rendering, he compared it to the original repeatedly to get it exactly right.

There's something to be learned here. When striving for realism, copy it. Take every possible measurement. Look at something from every conceivable angle and re-create it. If you're modeling a road, get outside and measure the one outside your house, and if that's not the one you're building, search for a similar road.

There's nothing as satisfying to me as creating something in LightWave and having someone ask "Is that a picture?" If you can fool them, you've done your job.

Of course, I'm not saying that everything you do should be photorealistic. The flying Listerine bottles on TV don't look realistic. It's apparent that they are computer generated, but can you imagine how strange those commercials would be if they did look real?

see Editor's Message, page 11

TABLE OF CONTENTS

LIGHTWAVEPRO
AN AVID PUBLICATIONS NEWSLETTER

Editor	John Gross
Managing Editor	Jim Plant
Editorial/Traffic Coordinator	Douglas Carey
Art Director	Helga Nahapetian Taylor
Editorial Intern	Joan Burke
Contributing Writers	Grant Boucher
	John F.K. Parenteau
	Mojo
	Greg Teegarden
	Mark Thompson
Publisher	Michael D. Kornet
Owner	Jim Plant
Cover Image Composed by	Greg Teegarden
..Image courtesy Amblin Imaging and Universal Television	

Editorial Offices: AVID Publications
273 N. Mathilda Avenue
Sunnyvale, CA 94086
Telephone (408) 774-6770
Fax (408) 774-6783

John Gross can be reached electronically at:
jgross @ netcom.com
71740,2357 (CIS)
Bubastis (AOL)

Printed in the USA© 1994 AVID Publications

Are you interested in writing for LIGHTWAVEPRO or submitting images? If so, contact us at our offices or electronically.

AVID Publications, its employees or freelancers are not responsible for any injury or property damage resulting from the application of any information in LIGHTWAVEPRO.

LIGHTWAVEPRO (ISSN 1076-7819); (Vol. 2, No. 7) is published monthly by AVID Publications, 273 N. Mathilda Ave., Sunnyvale, CA 94086-4830. A one-year subscription (12 issues) in the U.S. and its possessions is $72 (U.S.); Canada/Mexico, $84 (U.S.); Overseas, $108 (U.S.). Allow 4 to 6 weeks for first issue to arrive. Second-class postage rate paid at Sunnyvale, CA and additional mailing offices. POSTMASTER: Send address changes to LIGHTWAVEPRO, 273 N. Mathilda Ave., Sunnyvale, CA 94086-4830.

Letting it Rain on Your Parade

by Mark Thompson

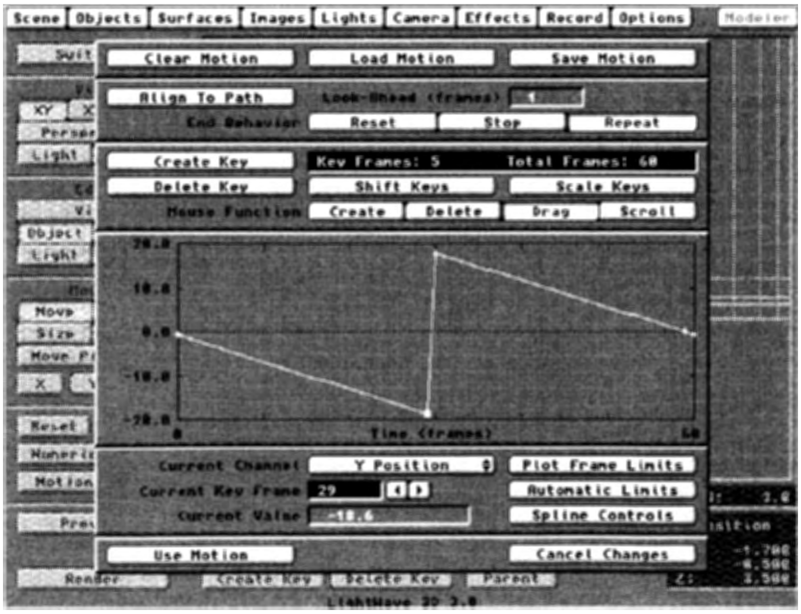
A couple years ago, I did a short animated film with the intent of showing off some of the great capabilities of LightWave. One of the film's most distinguished effects was a simulation of a rain storm. To this day, it remains the one effect that I get asked about the most. Since the theme of this issue of *LWPRO* is real world phenomena, I will describe how this most requested effect was accomplished.

There are two ways of simulating rain in a scene but perhaps the most realistic is through the use of particles or single point polygons. But rather than simply passing a sheet of particles in front of the camera, you want to use a full volume of particles which occupy a fair amount of your scene. This method has the two distinct advantages of reacting more realistically to lighting and allowing the camera to dolly though the rain.

Don't Forget Your Umbrella

The effect is accomplished in two parts: simulating the rain itself and the water on the ground. Simulating rain is simple. The mildly complex part of the task is ensuring that you have a continuous downfall of rain throughout the duration of the scene without modeling a column of particles 10 miles high. To do this, use two volumes of rain stacked on top of one another, rather than just one. Each volume should be large enough to entirely fill the camera view. Both volumes are given a downward path and move together in unison. But as the first volume passes below the ground plane, it is whisked atop the second volume and repeats its downward motion. The net effect is that there is always a full volume of falling rain within the camera view.

As an example, take the rain I used in my film. The particle volume was 18.6 meters tall in the Y dimension. Table 1 summarizes the keys used for the two particle objects. Also, Figures 1 and 2 show the motion graphs. This yields rain that falls at 18.6 meters/sec or approximately 40 mph. Whether or not this is physically accurate, it looked right in my scenes. There are a couple things to note. Since the rain should fall at a constant rate, use linear interpolation for the keyframes (splines would cause a minor rate fluctuation). Also, don't forget to enable Repeat for both motions.



Look-Ahead (frames) 1

End Behavior: Repeat

Key Frames: 5 Total Frames: 60

Current Channel: Y Position

Current Key Frame: 29

Current Value: -18.6

Figure 1

Table 1: Rain Objects Key Frames

Key	Rain #1	Rain #2
0	-0.5	18.1
29	-18.6	0.0
30	18.1	-0.5
59	0.12	-18.48
60	-0.5	18.1

When it Rains, it Pours

Of course, for the rain to look right, you must enable **Particle Blur** (Camera panel). The default value of 50 percent is fine. But a problem arises with Blur enabled. Since the rain repeats its motion and must snap back above the camera view after it passes below ground level, it leaves a large visible streak *[Editor's note: This visible streak is a known bug in LightWave that may be fixed with the latest version]*. In the example above, this streak occurs at frame 30 for particle volume 1 and at frame 60 for particle volume 2. To make the particles disappear for these individual frames, use Object dissolve. Tables 2 and 3 illustrate the values used in the example for the two dissolve envelopes. As with the motions above, select Linear for the keys and enable Repeat. Note that for Rain #1, keys are added for frames 31 and 60 to ensure a full 60-frame cycle.

Table 2: Rain #1 Dissolve Envelope

Key	Rain #1
0	0%
29	0%
30	100%
31	0%
60	0%

Table 3: Rain #2 Dissolve Envelope

Key	Rain #2
0	0%
59	0%
60	100%

Next, make these particles look more like rain when rendered. One thing to note about rain, it is for the most part invisible. Only when lit appropriately will the glinting streaks of falling water be seen. To accomplish this, use a very high specular value with 100 percent transparency. This will cause the rain to be visible when illuminated from a variety of angles, but largely invisible in ambient light. Table 4 has a typical set of attributes for a rain surface. Depending on the lighting conditions and application, you may wish to experiment with these values. Note that both diffuse and specular have values above 100 percent. LightWave 3.1 allows these attributes to be set up to 200 percent while previous versions had no such limit.

Table 4: Rain Attributes

Color	212,233,255
Luminosity	0%
Diffuse	150%
Specular	200%
Glossiness	Low
Reflectivity	0%
Transparency	100%
Edges	Normal

Don't Jump in the Puddles

The first half of the effect is now complete. The other part of the effect requires the simulation of the splashes of water that the rain would create. The best way to achieve this is through bump mapping a highly specular surface. The bump map is simply an animated brushmap that is tiled across the surface. Not many frames are needed for this sequence and it need not be very large. While my example uses only seven frames at 320x200, it wouldn't hurt to double both the resolution and the number of frames. This helps to make the splashes look a little more random.

There are a few ways to create this short image sequence. One method would be to use the Wave operator in MorphPlus. While this would generate very nice rain droplets that would spread without any manual painting or animating, they are a little too perfect and wouldn't look quite as natural as hand-painted splashes.

I chose to create the "splash-maps" in DPaint rather than MorphPlus. Start by creating about seven frames of animation in DPaint. All frames should be cleared to either white or black. Then, with a heavy brush, add several concentric gray circles, one per frame, centered around the same point. The circles should grow in size for each frame and the first circle should practically be a dot. Then use the various painting tools to smooth it out and muck it up a bit. I used outline, blur, smear and smooth. Use DPaint to sequence through the frames to see how they look.

Once satisfied, pick up the splash rings as an AnimBrush and save it. Then clear the anim and set the number of frames to something like 14. Now use the AnimBrush to randomly place splash sequences all over the image. Placement in time and position should be as random as possible. Any splashes on the edge must be precisely duplicated on the opposite edge so that the image seamlessly wraps on itself. As before, play back the sequence to see how it looks. When you are happy with it, save it out as a sequence of frames, not an animation since LightWave will not load anim files. See Figure 3 for an example of one image in my sequence.

With a sequenced "splash-map" finished, creating the rain puddles is simple. There are two options. Either directly assign the attributes to your ground surface, or create a separate "splash-puddle" object. Having a separate puddle object gives you greater versatility with the attributes for the underlying surface, but adds the rendering time penalty of a large transparent surface (which can be fairly significant). In the Stormy Night image (color pages), the splash was applied directly to the street while a transparent puddle object was used on top of the sidewalk.

Adding the splash is a trivial matter of applying a Y-axis planar bump map using the splash-map sequence. Size it so that the largest splash rings appear to be approximately four inches (100mm) in diameter. To make the splashes visible, you will need a large bump amplitude. I have used between 300-500 percent. The other pertinent attributes used in the example are given below in Table 5.

The Sidewalk Puddles surface is more completely defined because it is a separate object. Note that you could also create small irregular shaped puddle objects, but it may look a little funny around the edges where the bump texture gets cut off. Table 6 gives the information for the sidewalk puddle.

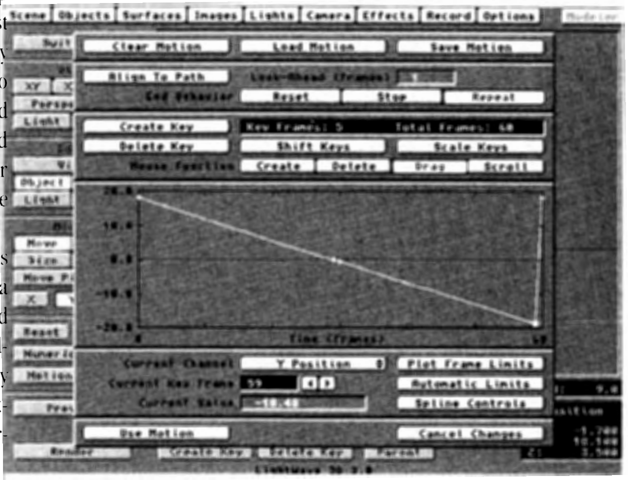


Figure 2

Table 5: Street Puddles

Specularity	150%
Glossiness	Medium
Bump Amplitude	300%

Table 6: Sidewalk Puddles

Color	200,200,200
Diffuse	60%
Specularity	150%
Glossiness	Medium
Reflectivity	0%
Transparency	70%
Bump Amplitude	400%

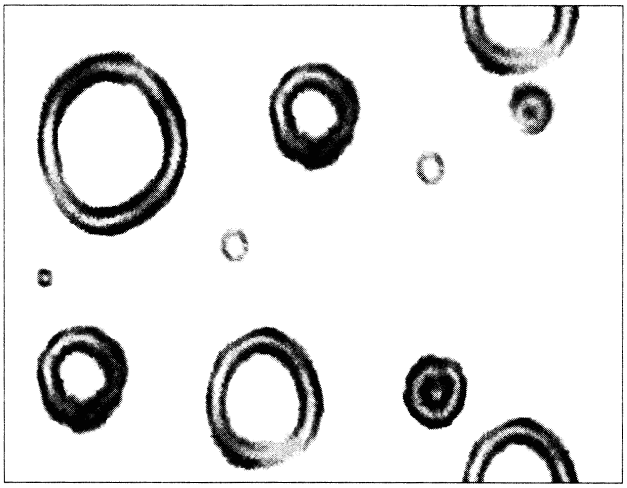


Figure 3

A Dark and Stormy Night

An added enhancement to the rain simulation is a little lightning. Just flashing a white light source for a frame or two adds a nice touch. You may use an existing light source, but it is probably easier and more convenient to add a new one. Make the light directional and all white with an intensity of zero percent. Then create an intensity envelope with one to three intensity spikes at the frames you want the lightning to strike. For example, over a six frame period, you might assign intensities of 0, 600, 70, 400, 30, and 0 percent respectively. Change the values and number of intensity spikes around for each strike. You may also want to slightly change the light sources' direction for each strike (since the flashes shouldn't always be coming from the same spot). Ideally, the lighting flash light source should be a shadow casting source, but chances are, for the short duration it is in effect, the lack of proper shadowing may be insignificant.

One final tip: When filling your scene with rain particles, it may be more convenient to break the volume up into smaller X-Z chunks rather than using two particle volumes (top and bottom) that fill the whole scene. Just set up the motion for the first two smaller rain volumes, clone them, and parent the clones to the originals. Then move the clones to fill in the rest of the scene. Similarly, this technique may be used to make a heavier rainfall. Just clone and offset a little bit from the original.

By following the steps outlined above, you can create a truly wonderful simulation of a rain storm. While you get a feel for the result in the Stormy Night image, this is an effect that really shines when animated. I should add that if you own Sparks, you could use it to generate the falling rain rather than setting up motions on particle volumes. However, you may exceed your object limit with that method. Best of luck, and let it pour!

Note: The film in which this rain effect was first used, "One Stormy Night with Fred Floaty," is available on Amiga World's "Amigamations" video tape. Contact Amiga World at (800) 441-4403.

LWP

Mark Thompson is president of Radiant Image Productions, a 3D animation and special effects production house. Send questions or comments to:
Radiant Image Productions
51 Derry Street
Merrimack, NH 03054
or e-mail to mark@westford.ccur.com.

Prometheus' Laboratory

by Grant Boucher

Sooner or later, you are going to start playing with those little buttons marked **T** next to your surface settings. While many of you use LightWave without getting deep into either modeling or surfacing, I am sure you've watched an episode of *seaQuest* or *Babylon 5* and asked yourself, "How in the world did they do that with LightWave?" I've had a lot of fellow LightWave users ask whether I used other programs to achieve high-end effects like fire, lava, plasma, dissolving spaceships, dripping mud, etc. In general, the answer is no. Everything you see on any episode of *seaQuest* is duplicatable with LightWave.

Our beta version of LightWave has some minor (and major) improvements, but rarely are those required to achieve a given effect. Usually they just make our lives easier as animators or give us new shortcuts to replace effects achieved through previously more complicated means. LightWave has had the core tools for such things since its earliest versions, and it's a major reason for its abundant use in the broadcast special effects market these days.

In my article on the plasma effect in a previous issue of *LWPRO* (Frankenstein's Lab, Issue 5), I tried to give you some logical building blocks for working with LightWave so that you could recognize when the techniques used in that exercise (i.e. sequential image mapped transparency and shaping surfaces with morphing and scaling) could be useful for similar effects such as lightning and lasers.

This article is intended to be another in a series on what I like to call the 'Zen' of LightWave procedurals. I want to continue to give you an idea into the thought process behind high-end special effects, so that the next image I see from you (either on the air, in print, or on your demo reel) shows me something I haven't seen before.

Chaos and LightWave

By now, I am sure you've heard of the **Fractal Noise** procedural lurking beneath most surface and object settings of LightWave. In fact, I used Fractal Noise with displacement mapping to create a rippling flag in the premiere issue of *LWPRO*. We are going to use almost identical settings and techniques to create fire.

Fractal Noise is a mathematical function which generates random looking variations between two settings. For example, when Fractal Noise is used as a surface color, you select two colors to randomly vary between. To make a diseased apple, you might make the main color red, and the Fractal Noise texture color brown or green. The **Texture Size** determines how small or large the pattern of variation will be, while the **Contrast** setting determines how smoothly the two disparate colors blend together. Contrast values lower than 1.0 become softer and subtler, while Contrast values greater than 1.0 become sharper and more defined. Fractal Noise can be used in all types of LightWave surface mapping, as well as displacement mapping (as in the case of our flag) and clip mapping (i.e.: if you wanted sharp random holes to appear in an object).

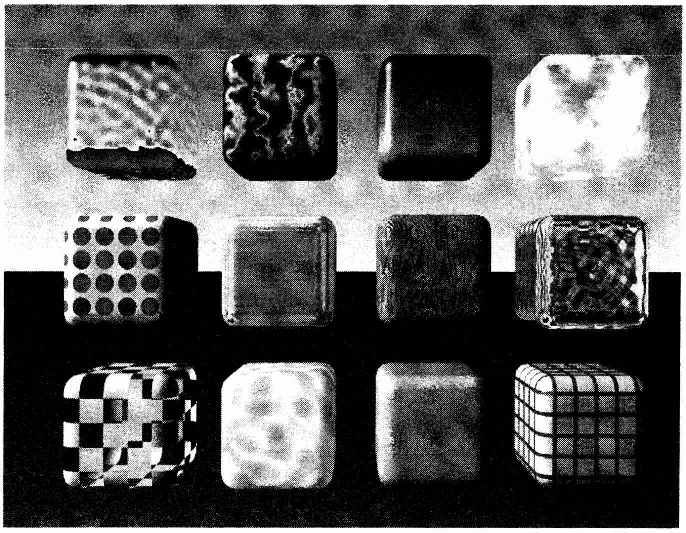


Figure 1

For example, load the Texture Examples scene that came with your version of LightWave and render the first frame (Figure 1). I'm going to go through these individually with a short sentence on each to give you an idea of the power and versatility of LightWave's built-in procedurals (especially Fractal Noise).

Starting from the upper left, the Rippling Chrome cube is simply the **Ripples** texture applied as a bump map to a surface with an environment reflection.

The Marble Cube is an example of a whitish Marble texture applied to a blackish-grey cube, purely as a surface color texture.

The Color Gradient Cube uses the **Grid** texture under Surface color to create a purple line on a blue-colored cube. **Texture Falloff** in the Y direction causes the purple line to fade away the further it is from the **Texture Center** (in this case, the bottom of the cube). Once the Grid effect reaches 100 percent **Falloff**, the grid is gone completely and leaves only the native surface color behind.

The Cloud Cube uses Fractal Noise as a transparency map. The surface color of gray could be anything we wanted, as the fractal pattern only concerns the opacity, or transparency of the surface.

The Dotted Cube used the **Dots** procedural as a surface color map.

The Brushed Metal Cube uses Fractal Noise again, but this time as a bump map (**Fractal Bumps**). The surface of the object uses the Fractal Reflections image map that comes with LightWave as a reflection image. Notice that the Texture Size on the Fractal Noise bump map is much smaller in the Y direction than in the X and Z directions. This causes the fractal pattern to be stretched out across the faces of the cube, giving us the appearance of brushed metal.

The Wood Cube uses a sample of the **wood** procedural as a diffuse map.

The Rippling Gold Cube is identical to the Rippling Chrome Cube above, except that a surface color of gold has been applied to all reflections by turning the **Color Highlights** option on.

The Checkerboard Cube uses the **Checkerboard** texture as a transparency map. Notice how easy it is to put

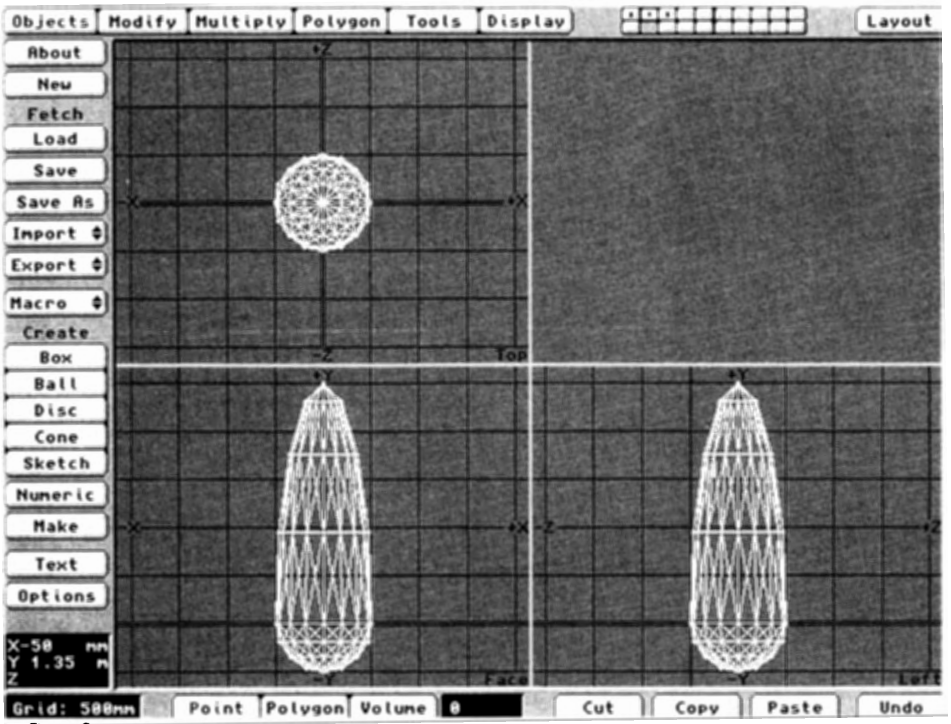


Figure 2

clean holes into objects without using the modeler and without increasing the number of polygons.

The Underwater Cube uses the **Underwater** texture as a surface texture with colors of white on blue to create that "bottom of the swimming pool" effect.

The Orange Peel Cube uses Fractal Noise as a bump map again, but this time with a small Texture Size to give a grainy feel to the surface. The orange surface color lets you know this is the surface of an orange, but the same settings can be used for sand and dirt effects.

Finally, the Grid Cube uses the **Grid** texture as a surface map to place grids on the cube.

Clouds and More

So what does all of this have to do with fire? Take another look at that Cloud Cube. If you applied that surface to a hemisphere you made in Modeler and placed your camera within it in the Layout, you could have a sky full of clouds. If you applied that surface to a Modeler cone and shrunk the texture size a little, you'd have a steam vent, aerosol spray, or torpedo trail. If you changed that same cone's surface color from white to black or gray, you'd have a smoke trail or volcanic plume. If you make that same object's surface color a fractal mixture of red and yellow (with shades of orange in between), you would get something that begins to look like fire.

Dissecting Mother Nature

So that is the essence of the fire texture, but like most good things, it's not going to be that easy. First, let's begin tackling fire by taking it apart, in LightWave terms.

Shape —A candle flame could be cylindrical, conical, or elliptical (i.e. egg-shaped). In either case, it is an easy task for Modeler. If we were going to place fire on a fireplace log, we might create an elongated cylinder or maybe a couple of flat polygons, or sheets of polygons might do the trick. Since we can't really make solid fire, we have to fake it a little.

In Modeler, select the **Ball** function in the Objects menu. Immediately select **Numeric (n)** and chose **OK** to accept the default settings. Press the **Return** key or select **Make** to create the sphere.

Next, select only the top four rows of points (i.e. any point with a Y value greater than zero) and proceed to the **Modify** menu. Select **Stretch** and the **Numeric** option to bring up the Stretch numeric requester. Stretch the selected points by a factor of five in the Y direction, leaving all other values at the default. After stretching, you should have an egg-shaped object that loosely resembles a candle flame (Figure 2).

The object will be much larger than a real candle flame, but I wanted to make the modeling easy as possible. You can scale the candle flame down in the layout according to your needs, without the need to adjust your texture sizes. I use this same fire object for large and small fires, so I am always scaling it up or down as needed. Save this object and enter Layout.

Color —There are many types of fire, with colors that range from bluish-white to yellowish-orange. Let's choose a more animated color scheme like red, orange and yellow. We'll choose a candle flame to be more specific. That will determine many of our other parameters.

Let's use a Fractal Noise pattern to range from red to yellow, and we'll make the Texture Size taller in the Y direction and narrower in the X and Z direction, since flames stretch upwards. Also, I have chosen to make the texture sizes small, which may resemble a bonfire more than a candle flame. You can easily adjust the Size values to meet your needs.

The following table gives the appropriate values to enter for surface color and texture:

Surface Color	255,0,0
Texture Type	Fractal Noise
Texture Size	0.1,0.3,0.1
Texture Center	0,0,0
World Coordinates	Off *
Texture Falloff	0,0,0
Texture Velocity	0,0.03,0.001*
Texture Color	255,255,0
Frequencies	3
Contrast	0.5

* Note that if you turn World Coordinates On, you can bank and pitch the candle flame and the flames will always travel upwards (i.e. in the +Y direction).

Luminosity—Flames do not receive shadows, nor do they shade according to the light sources around them. In LightWave terminology, that calls for 100 percent luminosity.

Transparency—You can see through parts of a flame, as well as the edges. Similarly, the flame fades away slowly as it travels away from the source until it disappears completely. It does not have any hard edges at all. To solve these many problems, we'll first use **Transparent Edges** to give our flame a soft-edged look. Many transparency-based effects work better with Transparent Edges turned on. For the holes, we'll turn to Fractal Noise again, but this time as a transparency map. With a falloff in the Y direction (starting from our texture center at the bottom of our flame Y=0), the flame will go from solid (0 percent transparency) at the bottom to invisible (100 percent transparency) at the top.

The following table gives transparency and texture values:

Transparency	100%
Texture Type	Fractal Noise
Texture Size	0.05,0.15,0.05
Texture Center	0,0,0
World Coordinates	Off (but see * note above)
Texture Falloff	85,40,85
Texture Velocity	0,0.03,0.001
Texture Value	0
Frequencies	3
Contrast	1.0

Movement—Flames travel upwards, so that calls for using LightWave's **Texture Velocity** functions. A +Y direction causes our fractal pattern to travel up and through our candle flame object. While not strictly necessary, a small amount of X and/or Z texture velocity will cause the fractal pattern to undulate slightly as it travels up the flame. This keeps our candle flame from looking too ordered and predictable.

Similarly, with LightWave's displacement mapping feature (Objects panel), our candle flame can waver to and fro and appear to have actual depth. The **Texture Velocity** settings for the **Fractal Bumps** displacement should be the same as those for our surface color and

see Prometheus' Laboratory, page 9

Beneath the Surface

by Greg Teegarden

The surface of the ocean is an ever-changing shape that lends itself well to the realm of the computer animator because of it's fluidity. LightWave has a number of tools that make the illusion of a body of water fairly simple to achieve. However, it's the amount of complexity and the factor of time spent rendering that will determine how successful and realistic your end results will be.

This month, I'm going to discuss how we at Amblin Imaging create the illusion of the underside of the ocean surface.

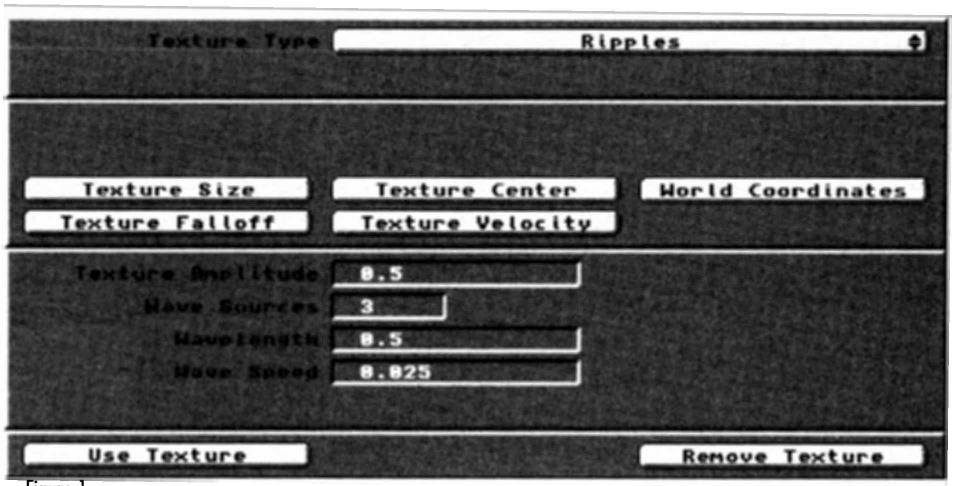
Come on In, the Water's Fine

There are basically two types of water surfacing employed to get the desired effect of an underwater environment. One is the actual creation of a water surface which is intended to be seen up close from its underside. The other is the caustic refraction of light traveling through the surface of the water and striking an underwater vehicle or creature. With the former, the latter is almost always employed, but usually not the other way around.

When I began working on *seaQuest*, it was determined that we would never see the underside of the ocean surface or even go above the surface. It was deemed too difficult and therefore unnecessary to even attempt it, but it was something I wanted to try nonetheless. One of the problems with attempting these types of tricks is that you will invariably create a situation that requires the use of raytracing, and this was often considered a bad word until fairly recently. With products on the market like the Raptor from DeskStation Technologies, raytracing becomes a practical possibility, especially within the constraints of a TV schedule and budget.

Usually the need for a certain look or situation is the catalyst for the discovery of a new technique, and realistic water was certainly one of them. The script for a *seaQuest* episode (titled "The Stinger")called for a "car race" underwater just below the surface of the ocean. Having posters on every wall showing what it looks like from six to 60 feet below the ocean surface, I had plenty of reference points.

The first thing I noticed was that the ocean surface resembled a crinkled piece of paper that had been unfolded. It is comprised mostly of smaller waves which I would consider calling wind "chop" riding on the backs of the larger waves. It is this "chop" that gives the ocean its unique look from its underside. This chop also creates all of the intricate caustic light patterns which seem to dance upon the surfaces of everything beneath it.



Texture Type: Ripples

Texture Amplitude: 0.5
Wave Sources: 3
Wavelength: 0.5
Wave Speed: 0.025

Figure 1

Dive In

In creating the water surface, the first thing I did was try and duplicate the larger waves which comprise the bulk of the movement across the ocean itself. This was done by creating a mesh of at least 20,000 polygons. This may sound like a lot, and it is, but to get something like a submarine underneath it and make it look believable requires at least a flat mesh of that size. I discovered that a mesh of around 24,000 suited my needs adequately. I should also note that the mesh is square in shape.

The next step was to create the larger or primary waves. Displacement mapping best suits this type of need, as it negates the need to morph, and LightWave has a good Ripple Displacement map. In the **Objects** panel you will find the Displacement Map area with the **T** button. Selecting this allows you to choose **Ripples** as your texture (Figure 1).

Since my mesh object was about 450 square meters , I played with the **Texture Size** values until I found one that gave me what I considered to be realistic looking waves, both in size and in separation from one another. I noticed that apart from the Texture Size button, a number of other options existed, including **Wave Sources**. The default value for this was three, but I found that the waves became more random as the value increased. With this in mind, I raised it to its maximum setting, which was 16.

Wavelength is another option that needed some adjustment. After a little experimentation, I found that a value of 16 seemed to work nicely here too.

Wave Speed is the distance, in meters per frame, that the wave travels across the mesh. I found that a setting of .025, the default, worked out nicely.

Finally, **Texture Amplitude** was set to .25, or that of a wave about 10 inches tall.

The Little Ones Get You

The next step was the creation of the secondary waves, which are most responsible for the caustic light refraction seen underwater. The best way to do this would have been to use another displacement map, however, you get only one per object in LightWave, and my object did not have enough complexity to support such a fine amount of actual displacement. The solution to this problem was to use a Bump Map. Bump maps work nicely because they actually bend raytraced light from an overhead source and concentrate it along the patterns on the map. Therefore, if you

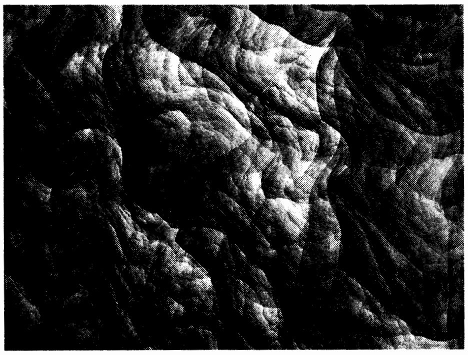


Figure 2

have a semi-transparent object that has swirling patterns on it and trace shadows through it, those same patterns will cast down onto the receiving surface. This works well for water caustic because everything in the scene below the wave surface gets illuminated by this random pattern of swirling light.This is exactly what happens in reality.

The trick to make this work for an animation is to use sequentially numbered images as the Bump Map. I have discovered that Apex Software's Forge program contains a nice caustics algorithm which can be animated over time. If I had this tool at my disposal earlier, I would have used it, but at the time all I had was several pieces of crumpled paper which I had scanned on a flatbed scanner.

As mentioned previously, the paper looked like the underside of the ocean surface when crumpled, so I used about five sheets of this as source images. I then used Photoshop to generate the in-between images, which creating a type of morph from one page to another until I had about 50 distinct RGB files (Figure 2). These were loaded into LightWave and used as the Bump Maps for the secondary waves riding the backs of the larger, displacement mapped primary waves. No texture velocity is needed on these RGBs, as the transition of one image to another is sufficient to carry the illusion of moving water caustics. As far as the size of the maps involved, obviously the bigger you can fit into RAM the better. The ones I used were the default 752x480 Toaster size, reduced in color depth to 8-bits per pixel in grayscale. That seemed to work well enough, but remember that lower resolution maps can tend to show signs of aliasing in the map itself, which destroys the illusion.

Pouring it all Together

We now have the basic elements needed to produce realistic looking water surfaces from the caustics and wave movement point of view, but what about the other settings? Water in itself does not have a color, but because it tends to absorb light in the red end of the spectrum, blue is the color that we see. I've tried many different surface settings and found that it looks best and most believable if the color settings shift towards blue. This may seem obvious, but since transparency is used to a high degree in order to allow light to get through, and **Additive** could be used with blue Zenith and Sky colors in your background gradient, one could make the water surface red and it would make no difference. In the end I opted not to use Additive and instead make the surface a dark blue color, with a light Sky background color. Transparency was set to 85 percent, with **Color Filter** turned on. Due to the Color Filter setting, which causes any light cast through a transparent object to take into account the color settings of the object in question, Additive was not used on the water surface. I wanted the light cast through to be blue in hue, and the Color Filter option achieved that nicely. The color of the light itself was that of the sun, shifted towards the yellow end of the spectrum. This creates a little bit of green here and there which makes everything look right.

Speaking of light, which settings work and which don't? In order to get the caustic effects, you must raytrace. There is no other realistic option I can think of, both aesthetically and practically. I boosted my **Light Intensity** to 200 percent, made the color a little yellow, and pointed it almost straight down. A distant light was used to create even saturation throughout the scene.

In the Surfaces panel, I found that because all of the polygons needed to be tripled in order to get a good displacement, **Transparent Edges** should be used to prevent any polygons from revealing their edges during the run of the animation. **Smoothing** and **Double Sided** were also used.

One final setting that had to be worked out was the reflectivity of the water surface itself. Ideally, **Trace Reflections** (Camera panel) should be used, but in the interest of time I opted not to use this feature and instead used an image of a scanned piece of crinkled tin foil as a reflection map. It worked well, with the reflectivity cranked up to 100 percent. I also used a 100 percent **Specularity** setting, with **High** selected under glossiness.

This all took about two weeks to figure out and test until I felt it looked good enough to pass for believable water. Included on the *LWPRO* disk this month is a sample scene file of the ocean surface I developed here at Amblin. I hope it helps to make your animations more believable and more fun to watch.

Greg Teegarden is an animator for Amblin Imaging. His work can be seen on seaQuest *and numerous commercials. He can be reached by e-mail on CompuServe at 71175,3025.*

Prometheus' Laboratory

continued from page 7

transparency map, but the **Texture Size** might be two or three times larger. Make sure you increase the number of polygons in your candle flame object to account for displacement mapping distortions. Otherwise your flame will crease sharply rather than flow smoothly. Also, start with a small **Texture Amplitude** such as .1.

Advanced Options

Diffusion & Double Sides—I selected **Double Sided** (Surfaces panel) to add additional layers of fire in our candle flame and to brighten up the color of the total object. A professional animator's trick when dealing with transparent objects that begin to disappear in scenes due to combination of transparency and transparent edges is to increase the Diffuse setting to greater than 100 percent. Since Diffuse is just a formula, you can bring up the amount of color applied to an object, before it is made transparent by going as high as 200 percentDiffuse (with LW 3.1). This tends to wash out your object, which can cause other problems but works great for our already 100 percent luminous, unshaded candle flame.

Additive —Many animators turn Additive on when using fire in their scenes. Additive causes the colors behind your Additive surface flame to be "added" to the color of the surface (up to 255,255,255). This can add a nice lightning effect for rocket engines and similar fire-like effects. Try it in a scene to see whether you like Additive on or not.

Glows—The Flame scene included on this month's *LWPRO* disk has three lens flares added to the candle flame to give a hot core and glowing edge to the candle flame. This gives the candle the appearance that it is actually generating light in your scene.

Flickering Light —For added realism, a point light has been placed at the center of the flame so that the flame lights the area around the candle. By ray-tracing shadows through the candle flame and perhaps adding a Light Intensity envelope, the light can appear to flicker, adding to the sense of realism in our scene.

Modifications and the Future of Fire

Some animators use the Marble texture as a luminosity or diffuse map to add yet another layer of random variation to their flames. Similarly, since flames come in many colors, from blue to white, you can expand on the ideas expressed in this article to develop any kind of flame or flame-like effect. Perhaps you can think of a reason to develop flames that reflect like gold or silver?

You might also want to take a look at Leo Martin's ProTextures Volume II. It has a great, seamless Fire image map which I've used extensively for fire and lava type effects.

With the rumor of more procedurals due for release with LightWave 4.0 and programs like Forge by Steve Worley's Apex Software available to LightWave animators, the possibilities for procedural animation techniques is expanding geometrically. If Fractal Noise can be used to create most of nature's wonders, imagine the possibilities with a hundred Fractal Noise-style procedures.

Grant Boucher is a supervising animator for Amblin Imaging and teaches seminars on LightWave uses when he has spare time (not too often). His main system hard drive at Amblin is named Prometheus.

Spotlight on Reality

by John Gross

You've seen spotlights in commercials and Movie of the Week intros, but how are they created?

Since LightWave light sources do not actually display 'beams' of light, one must build and surface any beam of light to be shown. The following step-by-step tutorial shows how to build and surface a beam of light that can be used in any animation.

The Model

- In Modeler, create a **Disc** using the **Numeric** requester (**Objects** panel) with the following values:

Sides	32
Bottom	0
Top	14
Center	0,0,7
Radii	4,4,7
Units	m

All other values remain at their default.

- Select the two end polygons of the tube and cut them out. Change the surface name to *OuterBeam* (**q**) for all of the remaining polygons.
- Copy the tube into another layer. Select **Stretch** (**Modify** panel) and input 0.6 for the X and Y Factor values while using the **Numeric** requester. Leave the other values at their defaults (make sure units is set to m) and press **OK**. Change the surface name to *InnerBeam* for all of these polygons.
- Select both layers as foreground layers. Choose **Taper 1** (Modify panel), then **Numeric** and select **Z** axis, - Sense and a factor of .05. Leave all other values at their default and make sure that the units is set to m. Click on the **Apply** button to taper the beam at the origin. The finished light beam should look like Figure 1.
- Save the two layers as a single object called *LightBeam* and load it into Layout.
- While in the **Objects** Panel, turn off all three shadow options: **Self Shadow**, **Cast Shadow** and **Receive Shadow** for the LightBeam object. You do not want a beam of light to cast or receive any shadows. Remember that these options apply only to the current scene and are not saved as part of the object.

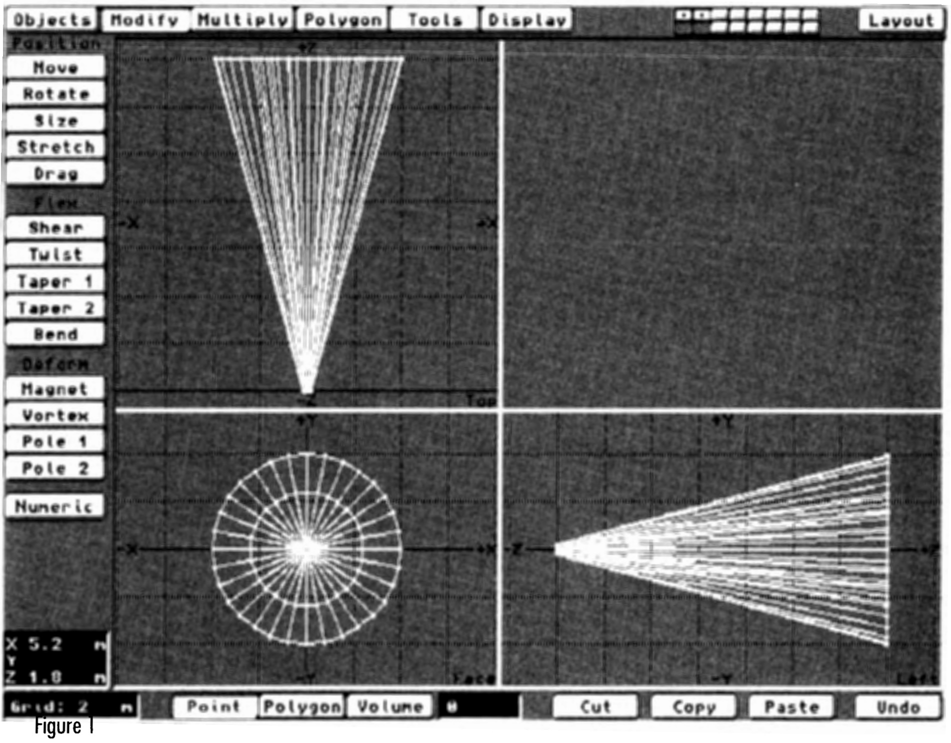


Figure 1

The Surfaces

Next, we will apply some surfacing to our light beam to give it a realistic look. One of the methods used to produce a convincing beam of light is to have the beam fade away at the wide end. We could use a **Transparency** texture falloff, but the problem there is that it will be a linear falloff. Beams of light do not fall off in a linear fashion. They are brighter at the light source, remain fairly bright for a distance and then fade off in an exponential manner. Often the end of the light beam will round off. For these reasons, I recommend that you do not use a transparency texture with a value entered for **Texture Falloff**, but rather an image to re-create the exponential falloff of a light beam.

Figure 2 shows a sample of an image (LightBeamFalloff.Brush) that works to ramp the transparency in a nonlinear fashion. It was created in ToasterPaint using the Range tools and then cropped in ImageFX to create a brush. The image is 224x480 in resolution. The white in the image will make the object completely transparent in those areas.

- Click on the Surfaces button to bring up the Surfaces Panel. You should have two surfaces for the LightBeam object, OuterBeam and InnerBeam. Select the following parameters for the two surfaces:

OuterBeam Surface

Surface Color	220,220,230
Surface Texture Map	
Texture Type	Fractal Noise
Texture Size	1.8,1.8,6.0
Texture Color	185,185,190
Frequencies	3
Contrast	1.0
World Coordinates	On
Luminosity	100%

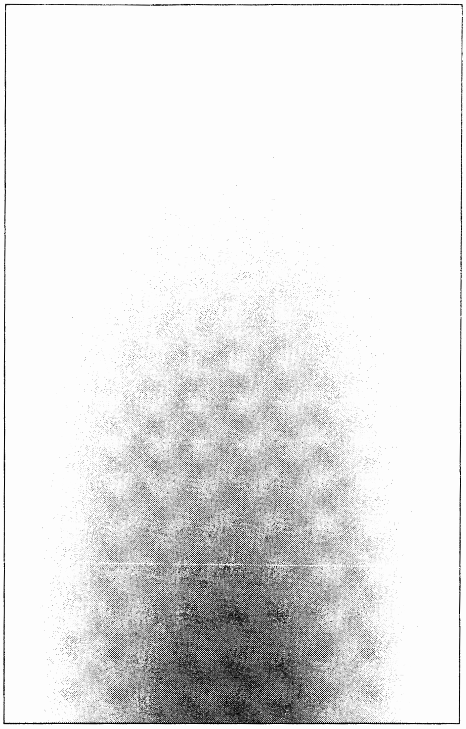


Figure 2

Diffuse	0%
Transparency Map	
Texture Type	Planar Image Map
Texture Image	LightBeamFalloff.Brush
Texture Axis	Y
Texture Size	8.0,8.0,14.1
Transparent Edges	On
Smoothing	On
Double Sided	On

InnerBeam Surface

Surface Color	225,225,255
Surface Texture Map	
Texture Type	Fractal Noise
Texture Size	1.4,1.4,5.0
Texture Color	190,190,200
Frequencies	3
Contrast	1.0
World Coordinates	On
Luminosity	100%
Diffuse	0%
Transparency Map	
Texture Type	Planar Image Map
Texture Image	LightBeamFalloff.Brush
Texture Axis	Y
Texture Size	4.8,4.8,14.1
Transparent Edges	On
Smoothing	On

- Rotate the beam or change the camera position so you can get a good look at the beam and do a test render (don't forget to set key frames).

The beam looks pretty good, but we're not quite done. If you need to shine the light on anything, you'll need to have an actual light source to do it with.

- From the Lights Panel, add a new light and give it the following parameters:

Spotlight Parameters

Light Name	BeamSpotLight
Light Color	240,240,250
Light Intensity	100%
Light Type	Spot
Intensity Falloff	5%
Cone Angle	18%
Edge Angle	5%

- Return to the Layout window and parent the BeamSpotLight to the LightBeam object. The spot light will jump into position inside of the LightBeam object. If you rotate the LightBeam object, the actual light source should also be rotating simultaneously.

If you are using LightWave 3.1, you'll notice that using these parameters will give you a cone of light that fits right inside of the modeled beam (take a second and look through the spot light in the Light view mode).

- For LightWave 3.1 users, adding a lens flare increases the realistic look of your light beam. Simply select the **Lens Flare** button in the **Lights** Panel for the spot light before clicking on the **Lens Flare Options** button and enter the following parameters:

Lens Flare

Flare Intensity	20%
Fade Off Screen	On
Fade Behind Objects	On
Fade in Fog	On
Central Glow	On
All other options	Off

This is a good starting place for your light beam. You may need to stretch the beam to fit your needs. If you stretch or size the beam in LightWave, there is no need to change any surface values. You can even size it out from 0 to full size in a few frames (try 5-8) to 'turn it on.'

Problem Solving

You may need to change the colors of the beam surfaces and the fractal noise that they travel through depending on the environment your beam is in. (Incidentally, **World Coordinates** is chosen for the **Fractal Noise** so the beams appear to travel through air particles as they move.)

If you need to create a different size or shaped beam in Modeler, however, make sure to adjust your surface values to compensate for the new object.

If you intend on traveling close to the light beam, you will most likely want to model some type of light fixture to place at the bottom of the beam. If this is the case, it is convenient to parent the LightBeam object to the light fixture and simply rotate the fixture around to move the beam.

To save rendering time, remove the InnerBeam and use only the OuterBeam. The fewer the transparent objects, the faster LightWave can render.

Finally, if you notice that the end of your light beam has edges showing, try increasing the **Z Texture Size** slightly to make the image map a bit larger to compensate.

LWP

John Gross is a supervising animator for Amblin Imaging and Editor of LIGHTWAVEPRO. *He can be reached electronically at jgross@netcom.com. On CompuServe, contact him or at 71740,2357.*

EDITOR'S MESSAGE

continued from page 3

Ever since version 1.0, LightWave has had the tools to create images that look real. Of course, each version gives you more and more tools, but the essentials have always been there.

What does all of this have to do with real world effects? Let's see if I can tie it in: Sometimes what gives a CGI image away is not the model, but the 'events' that are happening to that model. The explosions, the fire, the lighting, the rain—all of these computer generated effects can make or break a CGI shot. LightWave has the ability to create realistic looking effects with the tools you have.

This issue of *LWPRO* takes a look at some of those tools. You don't have to be a great artist to re-create the scenes reviewed in this issue. In fact, the effects in this issue can all be created without painting incredible image maps to add that photorealistic touch. Most of these effects are performed with procedural textures and simple image maps. Hopefully, these effects will start you on the road to creating your own impressive real world effects.

New on the LightWave Front

This issue arrives at the time (or slightly on the heels) of SIGGRAPH '94. There should be some exciting news in the LightWave arena that we'll be covering in future issues. Also look forward to coverage on the new standalone version of LightWave and related products in upcoming issues.

John Gross
Editor

Sunshine on Your Shoulders

by John F. K. Parenteau

Re-creating accurate lighting in a computer-generated environment is no simple task. In the world around us, there are often so many functions of the light that eventually reach our eyes that it is almost impossible to count them.

Observe the simplest lighting possible and you cannot count on your fingers the number of surfaces that either reflect, refract, generate or absorb light. In *LWPRO* Issue 5, Greg Teegarden and I touched on some basics, such as faking radiosity and the simplest lighting setups that all animators should consider. Though both discussions were complete in their own, each is just a component of the consideration involved in creating a convincing light scheme.

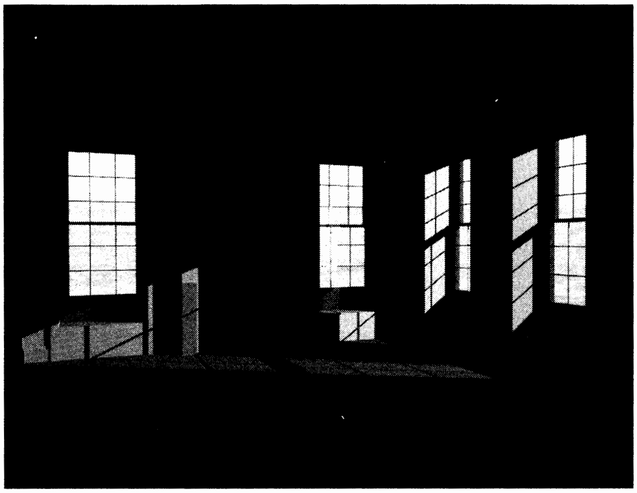


Figure 1

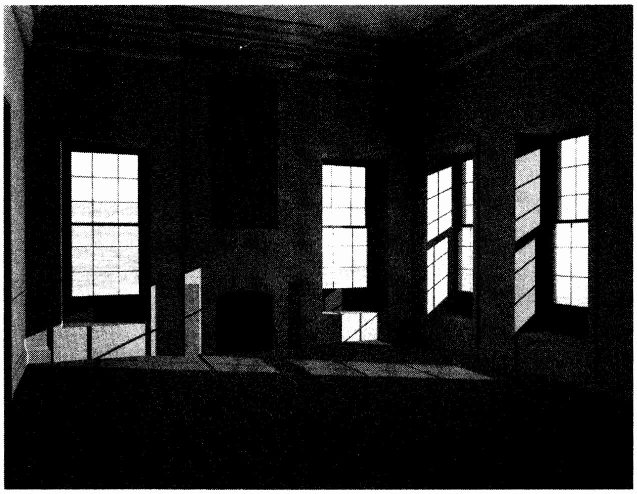


Figure 2

Lighting is Difficult

While teaching cinematography to film students, one of the most difficult concepts was the complexity and consideration involved in lighting a set. In almost any situation, a light source is relatively easy to come by. It is the job of the cinematographer to make the lighting attractive. Interiors, though difficult in their own way, often provide their own answers. Almost every interior set has a window, lamp or fireplace —and if there isn't one, it's a simple task to convince a production designer to put one in.

Exteriors, however, are a different story. If a character moves about in an interior set, it is often from one area lighting to another. This means there are certain areas on a set that are naturally lit. An example might be a person in a night library setting. Each table has a lamp to allow for reading and you can always motivate, or justify your light from there. An exterior night is more difficult. We have all seen many movies where the key source of light is the moon. How often is the moon really bright enough to illuminate a whole city block? Yet, if you think about it, there are always street lamps to create gothic pools of light, or doorways lit with neon. A lighting source can always be found or created. The task is to use those sources to create an appealing image.

Call it a Day

The following is one of the most incorrect statements that can be made about lighting:

"Of course, day exteriors are the easiest!"

In both real photography and CGI, a convincing day exterior is the most difficult lighting set-up to produce. It's easy to place one light up high, rename it 'Sun' and call it a day, but this will hardly look attractive and never look realistic.

The problem is never too little light, but rather too much light, which makes everything in the shot all too visible. Especially with CGI, it may take considerably more work to surface and model objects if they will be seen in such a revealing quantity of light. To create successful daylight, we must understand two important concepts: Color Temperature and Contrast.

All light has a color temperature. As you might extrapolate from those two words, the temperature of the light helps determine the color of the light. Standard daylight, usually considered when the sun is directly overhead on a clear day, is 5400 degrees Kelvin. This is a nominal figure and can vary from 6000 to 4600 and beyond, depending on the position of the sun in the sky, the materials in the air (i.e. water particles, clouds, smoke, etc.), and the amount of atmosphere between the measuring device and the sun. The higher on the visible spectrum of light, the bluer light appears.

To compare, tungsten light (similar to most light bulbs) is approximately 3200 degrees Kelvin. Existing low on the spectrum, this light tends to appear orange or amber. A candle has a very low color temperature and thus is very warm whereas a welding torch is extremely blue/white and has a very high color temperature.

Comparing daylight at different times of the day can explain why many exterior shoots, especially fashion photography, are shot either early in the morning or late afternoon. When the sun is low on the horizon, the amount of atmosphere and atmospheric debris, is greatly increased. The light tends to be softened and warmed (lower color temperature) by the added dif-

WARP ENGINE

The *Complete Acceleration Solution* for your Video Toaster/4000. The WarpEngine™ provides the Highest Speed 040 Acceleration available, with up to 128 MegaBytes of Local 040 Burst Memory™ and the Fastest SCSI-2 Fast HardDrive Controller Available! All of this WITHOUT using a single Zorro III Slot!

LightWave Users - Dramatically Increase your 3D Rendering Speed and reduce your load and save times Dramatically!

You do 3D Rendering for video production and each frame can take an hour or longer to render on your 4000/040. You're working on a 30 second commercial with 900 frames. It will take 38 days to render this 1 spot. Now you can install the WarpEngine 4040 into your rendering system and reduce your rendering time by up to a factor of 3. Now this same spot only takes about 12 days to render. Time is Money and you have just reduced your rendering time by up to 3. Don't wait - Get Warped NOW!

Take your Rendering system to Warp Speed with the WarpEngine4040 running at 40MHz for over 30MIPS!

How Fast it the WarpEngine™

Rendering the texture example included With LightWave:	A4000/040-25MHz	2 minutes, 40 seconds
	A4000Warp40MHz	under a minute - approximately to 3 Times Faster!
	A4000Warp28MHz	1 minute, 22 seconds - 2 Times Faster!

Diskspeed using 2.1Gig Barracuda Drive: Read from file: 9 Mbytes/sec CPU Available: 91% - up to 9 Times Faster

The 40MHz and 33MHz versions include the 040 Processors with Fan-Sink, four SIMM sockets and the NCR SCSI-2 Fast HardDrive Controller. The 40 MHz board requires 60ns Ram to work at its highest speed although you can insert wait states to use slower Ram.

The 28MHz WarpEngine™ comes without an 040 chip, four SIMM sockets and the NCR SCSI-2 Controller. This allows you to use the 040 from your A4000/040 and also the Memory from the Motherboard if you wish. This produces a 28MHz 040 Accelerator with 4 to 16MB of High Speed Local 040 Burst Memory™ and the fastest SCSI-2 HardDrive Controller available for the Amiga. This will double your Rendering Speed!

It is more cost effective to install a WarpEngine 4028 Accelerator if you have an Amiga 4000/040 than to buy a HardDrive controller/memory board or one of each which take up expansion slots and won't accelerate your Rendering System (Ram speed is actually slower*). **The WarpEngine 4028 is also upgradable to 33 or 40MHz!**

The WarpEngine™ uses standard 72 Pin SIMMs and allows the use of up to four different size SIMMs at the same time. GVP uses much more expensive custom memory modules only available from GVP.

The NCR SCSI-2 Fast 32Bit DMA Bus Master Processor makes it the Fastest HardDrive Controller available with a direct connection to the 68040 chip providing up to 10MB/s Transfers with 90 to 98% of the 68040 CPU Processing Power left available for Rendering!

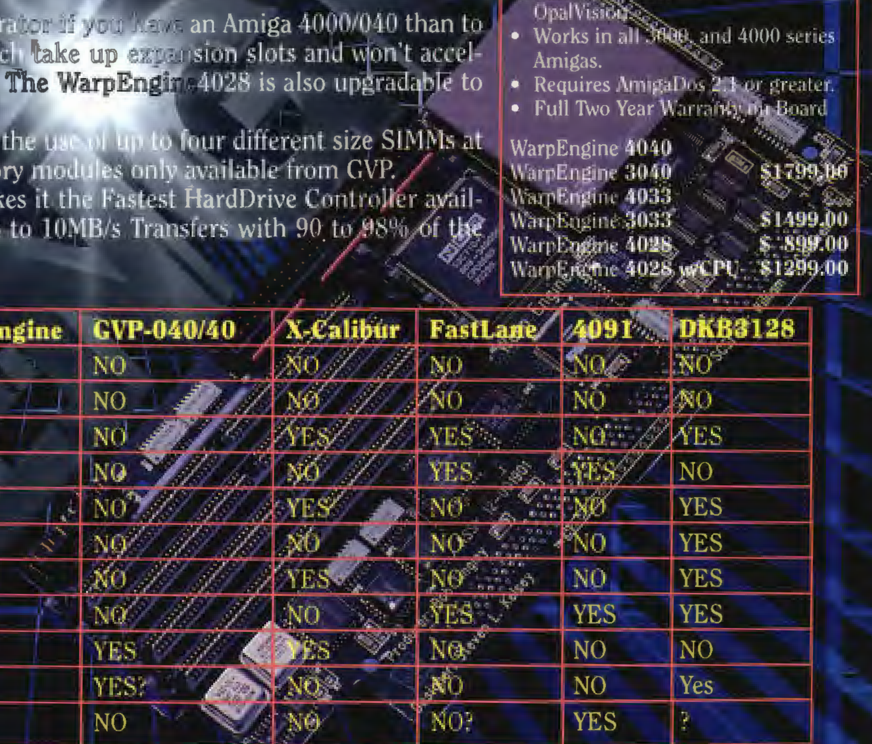
All WarpEngines are fully upgradable to 40MHz!

- Increase your Rendering Speed
- Speed up your Image manipulation using ADPro or ImageFX.
- Reduce Framestore load and save times dramatically.

Compatibility

- Compatible with the **Video Toaster**, Retina, Picasso, Emplant and OpalVision.
- Works in all 3000, and 4000 series Amigas.
- Requires AmigaDos 2.1 or greater.
- Full Two Year Warranty on Board

WarpEngine 4040	
WarpEngine 3040	$1799.00
WarpEngine 4033	
WarpEngine 3033	$1499.00
WarpEngine 4028	$ 899.00
WarpEngine 4028 w/CPU	$1299.00



Features	WarpEngine	GVP-040/40	X-Calibur	FastLane	4091	DKB3128
28MHz, 33MHz, 40MHz Versions	YES	NO	NO	NO	NO	NO
28MHz Upgradable to 33 or 40MHz	YES	NO	NO	NO	NO	NO
Expandable onboard to 128Megabytes	YES	NO	YES	YES	NO	YES
Built in SCSI-2 Fast Hard Disk Controller	YES	NO	NO	YES	YES	NO
Uses Industry Standard Amiga SIMM Modules	YES	NO	YES	NO	NO	YES
Uses any Combination of SIMMs	YES	NO	NO	NO	NO	YES
Allows use of the Memory from the Amiga	YES	NO	YES	NO	NO	YES
Uses a Zorro III slot	NO	NO	NO	YES	YES	YES
Memory Speed Much Faster than Amiga Ram	YES	YES	YES	NO	NO	NO
Works in Amiga 3000	YES	YES?	NO	NO	NO	Yes
Zorro III DMA or Buster Problems	NO	NO	NO	NO?	YES	?

*The speed of Ram installed on a Zorro III Memory board is about 30 to 40% slower than the Ram on the motherboard. [1]GVP uses custom memory modules that can cost up to 1.5 times as much as the standard memory used on WarpEngine. This memory can only be used on a GVP accel. Bd. [2]The X-Calibur requires you to install 2 SIMMs at a time. [3]The FastLane uses old 8Bit memory modules that must be installed 4 at a time.

MacroSystem Development, Inc.

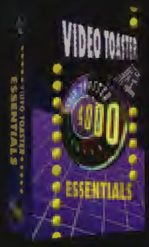
24282 Lynwood, Suite 201 • Novi, MI 48375 • Phone (810) 347-3332 • Fax (810) 347-6643

MAKE YOUR TOASTER SIZZLE

DESKTOP IMAGES INSTRUCTIONAL VIDEO TAPES

THE FAST AND EASY WAY TO LEARN THE VIDEO TOASTER

VIDEO TOASTER 4000 & 3.0



A step-by-step guide to the Video Toaster's digital video effects, set-up functions, ChromaFX and luminance keyer.

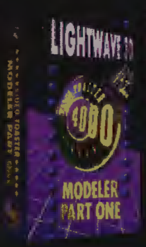
90 minutes

Complete instruction to ToasterCG's powerful text capabilities including font sizing, color brush loading, background & layering options.

100 minutes

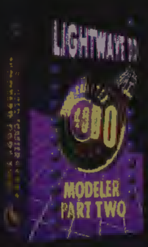
Discover the power of LightWave as basic 3D animation is explored. Learn scene creation, key framing, design and editing techniques for amazing animations.

120 minutes



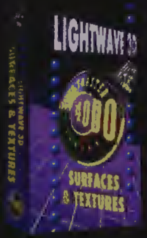
Create & modify basic shapes, making polygons with the freehand draw tool, using layers, creating text objects and assigning surfaces.

100 minutes



Change the shape and aspect of objects with modify tools, bending titles and logos, creating terrain using rail extrude, clone and lathe to create complex objects.

100 minutes



Professional surface and texture techniques, creating surfaces and increasing render speed by selecting the proper surfaces for your objects.

115 minutes

Twist, bend and deform objects in layout. Add incredible motion capabilities to your objects for realistic action effects.

100 minutes

Give your animation depth and realism with camera placement & effects, lighting selection, envelopes and animation techniques.

120 minutes

FEATURING

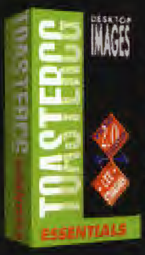
LEE STRANAHAN

Lee has taught thousands through his national seminars and published articles. In this professional video tape series Lee offers the tips & tricks that provide invaluable Toaster Power.

VIDEO TOASTER 2.0

A comprehensive guide to the Video Toaster's switcher and special effects, maximizing memory usage, ChromaFX transitions, and combining effects.

85 minutes



An easy step-by-step guide to the Video Toaster's character generator, font and palette controls, text file loading, and ToasterPaint CG backgrounds.

59 minutes

Discover the full potential of ToasterPaint's brush modes, warping & transparency controls, spare page feature, undo and redo functions and powerful keyboard equivalents.

62 minutes

Professional graphics techniques with multi-layered backgrounds, rub through and flood fill, textured ToasterCG titles, creating embossed borders and backgrounds.

59 minutes

Learn the basics of layout & renderer, loading objects, camera placement, motion envelopes, creating keyframes, quick rendering tips and animating to tape.

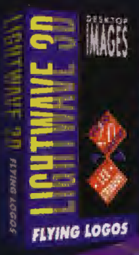
120 minutes

Create three dimensional objects using point manipulation, creating polygons, lathing & extruding, defining surfaces and creating complex objects.

90 minutes

Add texture and realism to objects by texture mapping applying images, using fractal noise, bump mapping, ray tracing, image capturing techniques & time saving tips.

90 minutes



Convert printed logos into 3D objects to create moving text graphics and flying logo animation, plus step-by-step instruction Pixel 3D Professional.

90 minutes

Each Tape is only **$49.95**

plus $4.95 shipping and handling

Call today for special package discounts

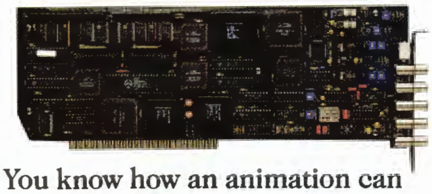
CALL TODAY TO ORDER:

1-800-377-1039

RAVE VIDEO P.O. Box 10908, Burbank, CA 91505
Phone (818) 841-8277 FAX (818) 841-8023

VideoToaster, ToasterCG, ToasterPaint and LightWave 3D are trademarks of NewTek, Inc. Desktop Images is a trademark of RAVE VIDEO

Save Your Animation From Being Eaten Alive.



You know how an animation can take on a life of its own. Sometimes it takes forever. Or it costs too much. Or a tape machine mistakes it for lunch.

The DPS Personal Animation Recorder™ solves these and other animation-production problems. For just $1,995, it gives you the reliability and capabilities of systems costing thousands more.

A plug-in AMIGA® card, the Personal Animation Recorder functions as a single-frame recording deck. With it, you can digitally record your animation onto a dedicated hard disk* and play it back in real time. Which means you can create 3-D animation without the expense and aggravation of tape decks. The Personal Animation Recorder will even genlock to your system.

Because the Personal Animation Recorder operates in a totally digital environment, you won't be bothered with the time base error, jitter, skipped frames, or botched edit points you encounter with traditional animation recorders.

Since your animation is recorded in a component digital 4:2:2 format, you can produce an infinite number of first-generation tape copies. Plus, the Personal Animation Recorder features outputs for true component analog video (Betacam®, MII®), composite and S-Video (Hi8®/S-VHS).

Rescue your productions from the jaws of traditional animation systems. Produce quality animation for a fraction of the usual cost with the DPS Personal Animation Recorder.

DIGITAL
PROCESSING SYSTEMS INC.
If you want to look your best

In the U.S. call (606) 371-5533 Fax: (606) 371-3729 In Canada call (416) 754-8090 Fax: (416) 754-7046

*Hard drive not included. DPS Personal Animation Recorder™ is a trademark of Digital Processing Systems, Inc. AMIGA® is a registered trademark of Commodore-Amiga, Inc. Hi8® and Betacam® are registered trademarks of Sony Corp. MII® is a registered trademark of Panasonic Broadcast.

Video Toaster® is out to amaze you. Again.

D2-quality Non-linear Editor

16 Bit CD-quality Audio

4 Input Switcher

35ns Character Generator

Video Paint System

3D Modeling & Animation System (with real-time playback)

Luminance Keyer

Hundreds of Digital Video Effects

Real Time Color Processor

Hundreds of PostScript Fonts

Introducing D2-quality, Non-linear editing for under $5,000.

What will we think of next?

The amazing Video Toaster just took another giant technological leap forward. Now, the world's first all-in-one, broadcast-quality, desktop video production studio can be turned into a sophisticated editing suite. Fact is, Video Toaster 4000 combined with Video Toaster Flyer delivers the same high-end editing capability you'd expect from a $50,000 digital deck—at a fraction of the cost. Check the list of features on the left side of this page, and start considering the possibilities.

To find out more, call now.

1-800-847-6111

Features, specifications, and prices subject to change without notice. Video Toaster is a registered trademark of NewTek, Inc. Toaster and Video Toaster Flyer are trademarks of NewTek, Inc. PostScript is a trademark of Adobe Systems, Inc. Video Toaster Flyer does not include hard drives. Complete Video Toaster 4000 Workstation with Video Toaster Flyer priced under $10,000. Video Toaster Flyer also sold separately for $4995. © NewTek, Inc. 1994

NewTek, Inc.
1200 S.W. Executive Dr.
Topeka, Kansas 66615

NEWTEK
INCORPORATED

fusion. In contrast, midday sun appears whiter (higher color temperature) and thus harsher. In addition, when the sun is directly overhead, shadows are much darker. Late or early light (often referred to as "Magic Hour") comes from a much more pleasing side angle.

Contrasting Situations

Contrast in daylight conditions plays an important part in creating believability. Many photographers use varying amounts of contrast to create the special look they are known for. Nature photographer Ansel Adams used angles of sunlight to create fantastic shadows or slashes of sunlight. Fashion photographer Herb Ritts shoots largely in daylight with black and white film, using the contrast inherent in color as his palette. Both photographers have manipulated contrast to communicate on film. In CGI, contrast will help bring realism to your image.

In Figure 1, I have created a simple scene in which we are looking from inside a room through a window to an exterior. First, I suggest using a spotlight rather than a distant light to act as a sun so that there is more control over its direction and cone angle. The keylight is raytraced with **Trace Shadows** (Camera panel) turned on and the **Ambient Intensity** (Lights panel) is set to zero. I like to use a slightly warm light, thus mimicking magic hour, for my Sunlight. A color value of 255,224,160 gives a great late afternoon look. Experiment with your settings for best effect.

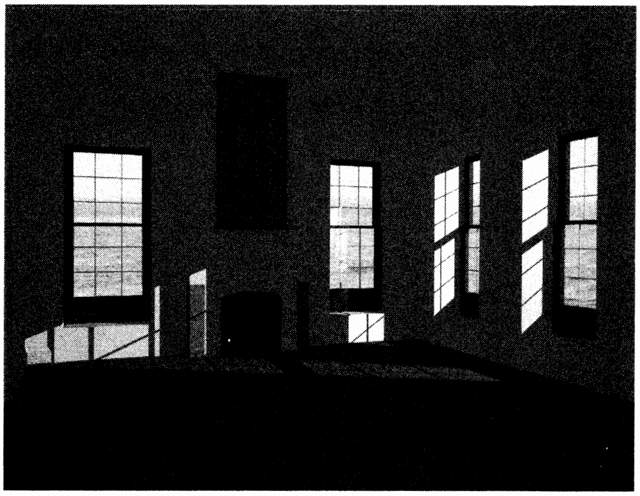


Figure 3

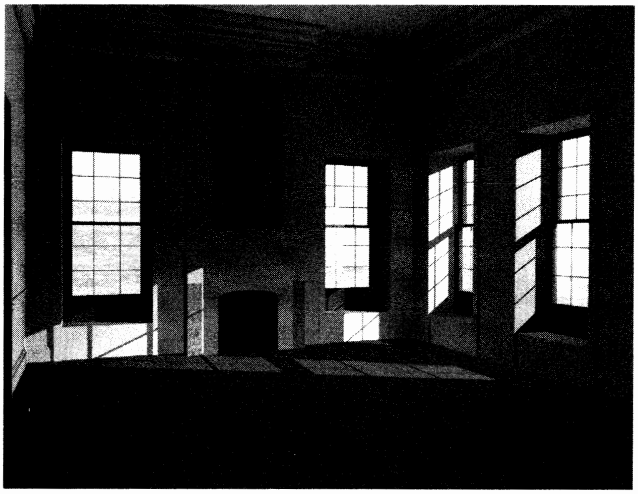


Figure 4

As in any CGI shot, compositing your computer generated image with a real video background immediately helps assist the shot's believability. In this case, I have used an ocean setting. Since there is no contrast in the background image, the sun must have been coming from directly behind the camera. In our setting, this will allow us to cheat our keylight to another location without it appearing too out of place. (If the sun in the background image was clearly on the left or right, casting shadows in a distinctive direction, we would be forced to mimic that direction with our light to maintain accuracy.) In this first setup, I have placed only a main keylight. Though the play of light into the room is appealing, the room appears far too contrasty.

Follow the Bouncing Light

All light has a tendency to bounce off of surfaces it strikes, thus creating an ambient light. With a small desk lamp in a night setting, the light from the lamp tends to fall off quickly due to lack of any great intensity. The sun, however, is very intense and will create a great deal of ambience.

In Figure 2, I have placed ambient 'source' lights in front of each window, disabling raytracing for these lights. This will mimic the ambience outside as sunlight bounces off the ground and sky, eventually bleeding into the room. The light cones are extremely wide (80 degrees) to give as nonspecific a source as possible. Light intensity is dependent on the amount of ambience you are seeking in the scene. In this case, I experimented and settled on a 35 percent intensity for both lights. Since ambient daylight tends to be a component of light bouncing off the sky and ground, it tends to be a cool, often bluish light.

In my CGI room, however, I have chosen to use the same color settings for my ambient lights as I did for my sunlight. This creates a much smoother transition between keylight and fill light. As you can see, this new light picks out the edges inside the room.

In Figure 3, I have deleted the ambience lights and increased the Ambient Intensity to 25 percent. The Ambient Intensity setting is highly unrealistic since it has no specific origination point. This means it strikes all surfaces from all sides. All surfaces are evenly filled from this non-specific light, and appear to have no edges. Rays of light, whether raytraced or not, all have a specific source. In Figure 2, ambience lights produce actual rays of light, striking the surfaces from one specific angle. Note the corners of the room. Dark areas (shadows) exist where the rays of light cannot reach. This creates definition of surfaces much more efficiently than an ambient intensity.

In Figure 4, I have added radiosity lights, estimating where the direct light would bounce off of surfaces onto other surfaces. Radiosity is a soft source that has fairly quick falloff. For this reason, I used settings of 17 percent intensity and 7.5 percent falloff. As mentioned in "Faking Radiosity," *LWPRO* Issue 5, these lights can be point lights placed just below the floor surface to avoid a hot spot. Radiosity from light striking side walls takes some testing. In this case, I placed radiosity lights just outside the windows to mimic light striking the window frame. This is a subtle effect and should not distract or compete with your main keylight. Do not raytrace these lights and make sure your floor surface is single sided, facing up. With the addition of well surfaced furniture and more realistic textures on the walls and floors, our scene will look downright real. This simple shot shows how basic consideration of how light falls can create a great looking image.

Another reason for using spots and points rather than distant lights is the ability to apply an **Intensity Falloff** (Lights panel). No matter how bright a light is, all light has a falloff value. Re-examine Figure 4 to see how falloff is best applied. Our keylight has a falloff limit well beyond our set. Though it has little effect on the amount of light striking visible surfaces, it does change the value of light over the distance it is throwing.

Composite This

Compositing into a realistic setting can be quite simple if you pay attention to a few basic rules. In my second setup (color pages), I am using the Humanoid object, sitting on a stool and placed in a live action tree environment. The tree image is a background image, and a plane has been placed under our subject with the same tree image **Front Projection** mapped on it (see image in the color pages).

Let's first place our keylight. If you note the shadows in the background and the shadow side on the trees, you can clearly see that the sun is low on the horizon on the left side of the frame. Careful analysis shows that the keylight should be set in a slightly backlight position, approximately 25-35 degrees off the ground, shooting light over the right shoulder of our subject. Though this accurately mimics the angle of the shadow in the background, the placement provides little if no light on our character's face. I have cheated the keylight around a bit to provide a little sunlight on the front side. If you now examine the shadows closely, our subject's shadow is not exactly matching. The cheat is acceptable, however, and hardly noticed. Due to the front projection mapped plane, I have chosen to use a Distant light source. Though I am unable to apply a falloff to this light, it will cover the plane evenly and thus allow the plane to match the background image better. The keylight will be the only light raytraced.

It took me a bit of analysis to establish the keylight color of 255,227,200. In examining the image, the sun-

see Sunshine on Your Shoulders, page 15

Pyromania
Creating Realistic Explosions

by Mojo

Babylon 5 is a new science fiction television program that deals primarily with humanity's social progress in the future. Since it is science fiction, viewers expect to see aliens, spaceships, strange planets and, without question, the occasional space battle.

Unfortunately, the high cost of traditional model-based special effects has meant that fewer space battles have made it onto the airwaves recently. Thanks to computer graphics, this trend has been reversed and *Babylon* 5 has proven that large scale space battles are once again indeed possible on a television budget.

Of all the elements needed to create intergalactic warfare, creating realistic explosions was unquestionably the most difficult. After much experimentation, I eventually developed an easy, fast-rendering, low-memory technique that produces surprisingly decent results.

Far Away, So Close

Distant explosions are simple. Just attach some expanding particles to a lens flare and voila, a little boom is created in the distance. Getting up close and personal is another matter. More detail is needed, specifically the flaming black-and-orange-organic-swirly thing we're so used to seeing blow up in the movies. Fractal Noise seems like the logical choice to somehow accomplish this, but the noise needs to travel outward from the center like a billowing cloud—not something this texture does on its own. A tip I learned on the *B5* pilot film provided the key to making this whole thing work.When given a velocity, Textures (such as Fractal Noise) will only move linearly—straight along the X, Y or Z axis. To get them to move in any other pattern requires you to morph the object it's applied to into a shape similar to the direction of desired movement (like the Saturn's rings tutorial in the latest Toaster manual). I knew a 3D ball probably wouldn't work, so I morphed a cone into a flat disk (Figures 1 and 2).

On this cone, a texture moving along the Y axis would move upwards from the base to the top point. If this point is dragged down to zero on the Y axis, the texture will still move along this path but now appear to travel from the outer edges of the "cone" to the center. If the texture is given a negative velocity, it will reverse course and seem to radiate from the center. Before you create this morph target in Modeler, you may want to delete the base polygon of the cone (highlighted in Figure 1); otherwise, this polygon will intersect with the ones dragged down and rendering errors will occur. Save the cone and its morph target and make sure you name each with different surfaces (I usually leave morph targets named default).

Layout

Once you've loaded both objects into Layout, leave the cone permanently morphed into the disk and dissolve the morph target completely out. Orient the disk so it is directly facing the camera. Since this is a flat effect, you want to be sure the disk is never at an angle or else its 2D nature will reveal itself.

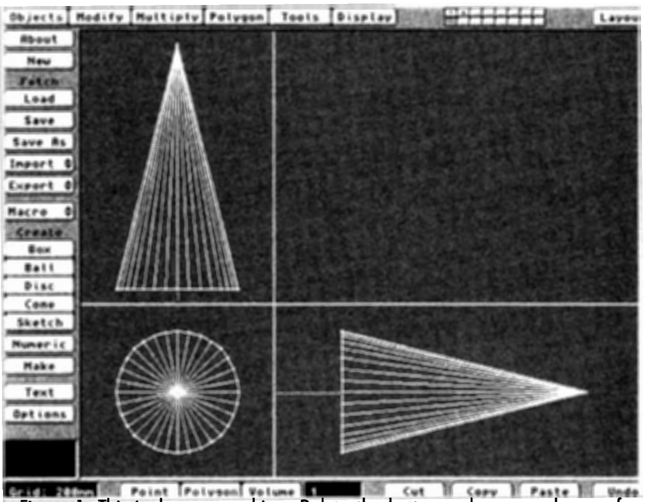


Figure 1: This is the source object. Delete the bottom polygon, apply a surface name and save it.

Surface it with a yellow-orange color and add black Fractal Noise with a negative velocity along the Y axis (on a 1 meter disk, I suggest approximately .1 **Texture Size** values). Before going any further, try to render a simple 30 frame test of the morphed disk against a black background. Seeing it animated before you clutter it with the other explosion elements shows what it does and probably includes a few ideas for other uses of this object.

Since few fireballs have hard edges, you'll need to create a circular transparency map for the disk which gradually dissolves its edges. As good as they are, I found that ToasterPaint's transparency tools didn't quite provide me with the results I was looking for. Instead, I rendered a white lens flare in LightWave against a black background and saved the RGB file. The soft falloff of a lens flare lent itself to a perfect map for this purpose. Just make sure the flare is small (so it actually dissipates to black on screen) and turn on negative image when you **Planar Image Map** it (**Y** axis) onto the disk. Render a test frame to be sure it looks right.

Spark of Life

You'll definitely want to make some particle objects to simulate flying sparks. These are a must if you want a realistic explosion and are probably the easiest part of the entire assembly. Since we're aiming for a two-part explosion, you should create two separate particle balls, perhaps each with different characteristics. I used the **Point Distribution** macro to create a ball of single point polygons that could be sized from zero to a much larger size in Layout. Figure 3 shows a skewed particle spray I created for a second point ball. By expanding the sparks outward toward the camera, they appear to move much faster and add a nice contrast to the first ball, which expands evenly in all directions. It might also be a good idea to create several surfaces within the points so you can make multi-colored sparks. Save these objects out and you're ready to earn your virtual pyrotechnician's license.

Some Assembly Required

Now it's time to put everything together and actually make something. First, load and clone your disk first, since you'll need two of them for the multi-part explosion effect. It might even be a good idea to use two separate cones with different surface names so you can apply varied surface settings (like alternate colors or texture speeds). Size the first one to zero and create a key at frame 1 for it. Size it up to almost fill the screen at around frame 5, then have it scaled down to about half its size by frame 30. It should look as if it expands rapidly and slowly dies down. Make sure you also set a similar dissolve envelope for it, so the explosion object fades away as it scales down.

Place your first particle ball in the center of the disk and keyframe its size at zero. Expand it to at least three or four times the size of the screen and key that near frame 20 so it coincides with the first disk object. You probably want these to dissolve out slowly, maybe beginning a few frames into the animation. The surfaces should have at least 100 percent Luminosity and be bright in color. Also, make sure you turn on particle blur (Camera panel) or else you won't see any sparks. Increasing the **Blur Length** might not be a bad idea, either.

For all these envelopes, especially motions and dissolves, use the spline controls frequently. The fine-tuning they provide can make all the difference in realism. Start them off quickly with a **Tension** of -1 and ease them out slowly with a **Tension** of +1.

The second stage of the explosion should more or less be a repeat of the first part, although offset by five or 10 frames. If you want the second part to be more violent, use more particles for the second wave of sparks and size up the disk larger —perhaps have it scale down and fade out over a few extra frames. Everything should be over by frame 30 or 45. Also be certain to move the second disk slightly in front of or behind the first; if they occupy the same space you will get rendering errors.

Last but not least, place a lens flare right in the middle of the object and ramp it up and down appropriately. You may want to stagger its envelope, having it die down after the first part and flare up again with the surge of the second stage.

The Secret of Comedy and Explosions

Proper timing of all your elements is the key to this effect. All the individual pieces may look great, but unless they are orchestrated properly it just will not seem right. You probably won't get it right the first time, so be prepared for multiple test renderings. Fortunately, render times shouldn't be too long unless you add even more disks for increased density in the gas cloud (multiple levels of transparency are a real render hog). Constant tweaking of the dissolve and motion and flare envelopes will inevitably help you tailor the explosion to your liking.

Don't be afraid to also play with the surface settings to help you get the desired effect. Changing the Texture Velocities and Sizes can make a big difference. Nothing here is set in stone; like everything else in 3D, bold experimentation will reap the best rewards.

On Disk: Included on this month's disk is everything you've just read about. Just load and render the provided scene and avoid doing any work yourself. Everything is attached to an explosion null object, so the entire assembly can be moved or even attached to another object. Also, note the disks are doubled up, which provides for a brighter explosion but much longer render times. Delete one of each for faster, yet still quality results.

LWP

Mojo has realized that nobody out there really knows anything about him. Though he has written bio after bio, Mojo has yet to print any factual information. Is he afraid of something? Is his past so sordid and depraved that even the smallest detail would result in his immediate prosecution? Young readers should understand that a career in 3D animation does not automatically dictate a life of deviance and seclusion. As in the case of Mojo, this only happens if you are extremely lucky.

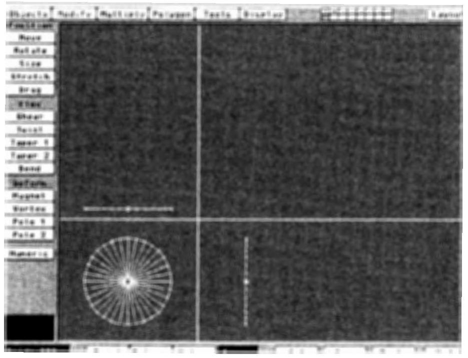


Figure 2: The top point of the cone is dragged down to create a flat disk. Give this the default surface setting and save it as your morph target. The animated texture will be forced to follow the contours of the disk.

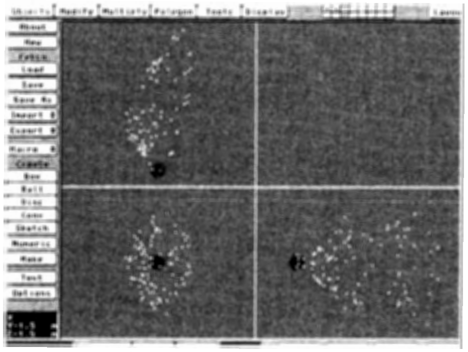


Figure 3: The background layer shows the original ball of particles. The foreground displays how they were stretched and tapered into a shape which will expand out into the camera.

Sunshine on Your Shoulders

continued from page 13

light appears extremely warm. This is actually not a component of the color temperature, but rather a function of the material on the ground. Dried up leaves and branches take on a deep brown color. As the slightly golden light of sunset strikes these surfaces, they reflect an amber hue. When choosing a light color, it is important to examine the entire frame. Note the light striking the leaves overhead. This light tends to be green due to the color of the leaves. In the distance, a white fence appears appropriately white. In noting all these variations, I came to the conclusion that the sunlight was still fairly high in color temperature. The value I have chosen is slightly yellow, which mimics a color temperature of around 4800 degrees Kelvin. Since I am using a front projection map on the plane below the character's feet, it is effected by the color of the light striking it. Using a slightly yellow light enables me to warm the character up a bit without discoloring the mapped plane under it. If you look closely, you can actually see the line where this warmer light is not matching the background image.

With our current lighting setup, we have established an extremely contrasty image inappropriate to the background map. For general fill light, matching sunlight bouncing off the sky in the distance, I have placed two point lights approximately two and a half meters from the front and the left of the shadowed side of our subject. These lights are the same color of the keylight placed just over the head of our character and have an intensity of 13 percent. A falloff of 1 percent should be applied to both lights.

To mimic the sunlight striking the ground around our subject, I have placed two point lights just below the ground plane. One is placed near the outstretched foot and another behind and to the right of our character. My plane is single sided, with the face pointing up, and neither light is raytraced. Using an Intensity of 35 percent and a Falloff of 13 percent gave the best results.

Finally, the efficiency of LightWave raytracing has provided a far too specific shadow in our scene. Adding some luminosity to our ground plane, while decreasing the diffuse (to prevent the plane from becoming brighter) will help dissipate the harshness of our shadow. A luminosity of 50 percent and diffuse of 75 percent in this case, best accomplishes this task.

Examining the final image of our new setup, the light striking our 3D element looks fairly appropriate. With better surfacing, however, the effect would be much more believable. Lighting for daylight is, as in any CGI environment, only as realistic as the CGI elements modeled, mapped and lit. Careful study of a background image, or careful planning for the look you wish to achieve will provide simple lighting answers for any setup. Just as in any real-world photography, attention to detail will pay the most dividends.

LWP

John F.K. Parenteau is a vice president of Amblin Imaging, where his experience as a cinematographer has helped add realism to the company's CGI effects.

Reader Speak

Subscribers' Questions and Comments

by John Gross

Well, the "reviews" are in and it doesn't look like everybody enjoyed issue No. 6. Here's some of the mail I've received.

Hi John,

I am a subscriber to both *Video Toaster User* and *LWPRO*. I enjoy *VTU* and find it highly insightful with its variety of columns and timely reviews. I paid for the *LWPRO* subscription because I am trying to achieve my dream of making a living from the activity I most enjoy: 3D Graphics. I anticipate the arrival of each month's *LWPRO* to hear the latest batch of tips, tricks, secrets and constructive suggestions from those artists whose work I have come to admire.

At $48 a subscription for 19 pages (including the cover and four pages of color photos), *LWPRO*'s info comes at a premium. As important as I found the reviews in April's *LWPRO* to be, I would much prefer to see these articles in *VTU*. I would not, however, object to an article about how to achieve a specific effect with Sparks or a tutorial on how to most efficiently combine several Power Macros with "X" to achieve some special effect.

Angel Freire
via Internet

Hi John,

I wanted to send you a short note about the current issue of *LIGHTWAVEPRO* (No. 6). I have been very pleased with all of the previous issues of *LWPRO* and I have found that it was well worth the subscription. However, I didn't feel that this issue quite lived up to that standard.

As a subscriber to *LWPRO*, I anxiously wait for each issue, expecting to learn new, important things about LightWave. An issue devoted to third-party software was a little disappointing. I expect that I will see that kind of information in *AmigaWorld* or *VTU* or even on the Internet or CIS.

Matt Mower
via CompuServe

Dear John,

I have been completely happy with *LIGHTWAVEPRO* since issue No. 1, regardless of the high introductory subscription fee. However, I am not satisfied with my new issue devoted entirely to product reviews. Are you running out of LightWave ideas?

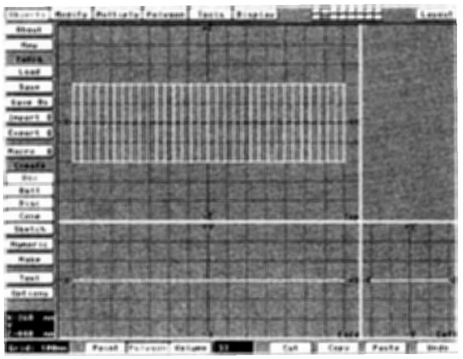


Figure 1

Please continue to provide issues filled with LightWave info. I wouldn't mind if reviews were included in regular issues. Tutorials for third-party products would also be acceptable. Please don't make this "review only" thing a habit.

Dan Kosmal
Albany, CA

In Response

It was entirely my idea to have an entire issue devoted to third-party products. I feel it is an important part of the information a competent LW artist needs in his/her arsenal. The reason the entire issue contained reviews is that I wanted to 'catch up' on the products that had been out for a little while.

Chances are, in future issues, there will not be a need to devote an entire issue to product reviews. However, I will continue to include reviews of products that are designed to work with LightWave in order to keep you up to date on what is out there and what can be beneficial or detrimental to your animation projects. Expect these reviews to be 'scattered' throughout the issues and not lumped all together.

I feel that, because *LWPRO* is a newsletter, the reviews can be a bit more beneficial than those seen in magazines. Not just because our turnaround time is not as lengthy (3-4 months for many magazines), but also because the reviews in *LWPRO* are conducted by LightWave professionals who provide the kind of information you need to make an informed decision about products you are considering buying (and not products you have already bought).

As a side note: I've been told by a few people that they feel the subscription price of *LWPRO* is too high. I'm not going to defend the price by saying that it can easily pay for itself with a few tips that speed up a project (Oops, I just did). Or that you could write it off on your taxes (Oops, there I go again).

You may not realize this, but in many cases, subscribers do not pay for most of the costs involved in producing a magazine, but rather the advertisers do. As a newsletter, however, *LWPRO* relies on its subscribers' loyalty to survive.

Regardless, I know cost is a relative thing in this case, but think about this: I subscribe to some newsletters that are well over $100 per year.

Before I get any complaints about wasting a column, let me answer several LightWave questions I received last month.

Dear John,

I sometimes get a message while trying to lathe an object in Modeler that says "Source contour contained no edges. Only points were generated." Is this a bug or am I doing something wrong?

Denny Stone
Downer's Grove, IL

In order for an object to be lathed, it must be an open contour in order to work. This means it must be composed of curves and/or open polygons arranged into a shape which is not closed on itself. For instance, a flat disc created with the disc tool will work, but a 3D disc will not.

Theoretically, the same should hold true for extruding as well, but I have noticed that with version 3.1 of Modeler, if you try to extrude a primitive built from the Objects panel, you will get the same message, but it will generate a duplicate of the object with all of its polygons facing in the opposite direction from the originals.

Dear John,

I have had a hard time getting ray-traced refractions to work properly. I am trying to re-create my dining room table (which is made out of glass), but when I render it with refractions on, it does not seem to refract light the same way the real table does. I saw a post on a local board that referred to a possible problem with refractions in LW. Is this an error in LightWave, and can it be corrected?

Francisco Négron
Tallahassee, FL

First of all, there was a bug in LightWave 3.0 with refraction. When many of the rendering calculations were changed to use floating point, one of the variables in refraction was not properly changed, which could lead to rendering errors. However, this bug was fixed in version 3.1.

It is possible that your dining table object wasn't constructed properly for ray tracing. Each surface that a ray passes through changes the index of refraction. To be perfectly accurate, the outward-facing polygons should have a surface with the refractive index of glass, and each of these polygons should be paired with an inward-facing polygon whose surface has the refractive index of air (or whatever material the ray would then be entering). This is quite simple to create in Modeler. Suppose you have a glass table with the surfaces named "table top" facing outwards. Simply copy this object into another layer, change the surface name to "table inner" (or something appropriate) and then press the **f** key or select the **Flip** button in the Polygon menu. This will flip all of the polygons inward.

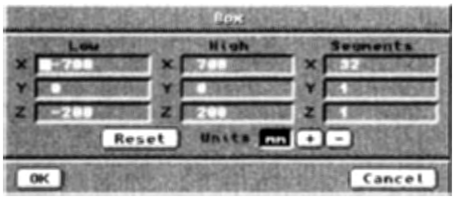
Now copy this object back into the original layer. You can then merge points to get rid of the duplicates. Save this object out and reload it into Layout. In the Surfaces panel, set your refractive indices for the appropriate surface names. With fully defined transitions, you should get accurate results.

You could also be experiencing one other problem. Version 3.1 has another bug in the refraction calculations which prevents rays from bending when exiting glass (in other words, when they pass through the inward-facing polygons that have a refractive index of 1.0). There is a simple method to avoid this, however, by giving the inward-facing polygons a refractive index slightly greater than 1.0.

Dear John,

I've read and re-read Grant Boucher's Lightning Ball article, but I don't understand this paragraph:

"If you build a lightning image box that has a lot of segments on the X direction (say 32 or 64), you can then morph that object into one that has been bent around the Z axis into a disc in LightWave's Object menu. The sticky texture can..."


Box

	Low		High		Segments
X	-700	X	700	X	32
Y	0	Y	0	Y	1
Z	-200	Z	200	Z	1

Figure 2

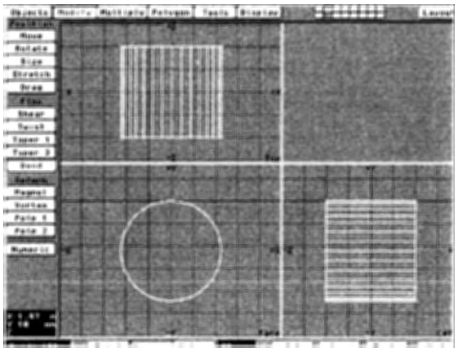


Figure 3

I don't understand this entire paragraph or what Grant is getting at. What is a segment? What is a sticky texture? Also, do you know what happened to the LightWave mailing list? It just stopped all of a sudden. Do you have any other suggestions on how I can get LightWave questions answered?

Doug (dnakakihara@BIX.com)
via Internet

A segment is simply a polygonal "cross section" of an object. For instance, Figure 1 shows a flat plane composed of 32 segments along the X direction. You can create such an object by using the **Numeric Box** tool and inputting a value greater than 1 for the **Segments** field (Figure 2). You can change the number of segments in any of Modeler's primitive objects by using the Numeric requester before you make the object. If you need to bend, twist or modify objects, they must be composed of many segments to do the bending or twisting along.

Grant was referring to creating such an object, saving it, and then using the **Bend** tool (Modify menu) to bend it back onto itself to create a rolled up section of polygons (Figure 3). This new object can then be saved as a morph target for the original flat object. This is where the "sticky texture" comes into play.

What Grant meant by this is to simply apply an image map to the flat plane of polygons (the morph source) and then morph that object into the rolled up object. There is no need to use an envelope. A value of 100 percent will be just fine. When an object is morphed, any texture applied to it "goes along for the ride" as if it is sticking to the surface. This is a great way to get image maps to appear properly wrapped on odd shaped objects.

Tip No. 1: When bending an object, first use the mouse to get the bend in the approximate position then **Undo** it. Next, select **Numeric** to bring up the Bend requester and look at the values. These values are those that you just performed with the mouse. Chances are they aren't going to be nice round numbers, so just change them so they are. For instance, if the **Angle** value is 179.432, change it to 180. If the **Direction** reads -87, change it to -90. When you select **Apply**, the modification will be perfect.

Tip No. 2: When creating a morph target, do not cut and paste the original object into another layer to modify it. Doing so changes the point order of the object and will cause a morph to take on the qualities of a rat's nest.

Instead, create the source object in one layer and save it, then modify it directly in that layer and save it out as your target object. You can place a copy of the original in another layer if you wish to use it as a background template for you modifications.

Tip No. 3: Your target object does not need to have any polygons. Since LightWave is only concerned with the point order when performing a morph, it doesn't care if there are polygons or not. Removing the polygons (**k**) before saving a morph target will help save memory, and you won't have to make your target invisible since points do not render in LightWave.

Finally, the Internet LightWave mail list that you are referring to was canceled because the sysop (Bob Lindabury) lost his Internet connection. A new list has been started and chances are you have been automatically subscribed. For those of you interested in signing on to the LightWave mail list, you can subscribe by sending e-mail to listserv@netcom.com and state in the body of the letter: subscribe lightwave-l *yourname*. You will begin receiving mail from users all over the world that is geared towards LightWave and Modeler. Posts by Allen Hastings and Stuart Ferguson and other LightWave professionals are not uncommon. LWP

What's on This Month's LIGHTWAVEPRO Disk?

This month's *LWPRO* disk includes sample scenes and objects covering all of the real world effects discussed in this issue. To order your *LWPRO* six-disk subscription, call Toll Free 1-800-322-2843; rates are: $30 U.S., $40 Canada/Mexico and $50 Overseas.

In the August Issue ...

In our next issue, we'll cover lip syncing in LightWave, a review of Apex Software's Forge program and several new columns. Also look for news from July's SIGGRAPH '94 convention in Orlando, Fla.

Animate like the PROs!

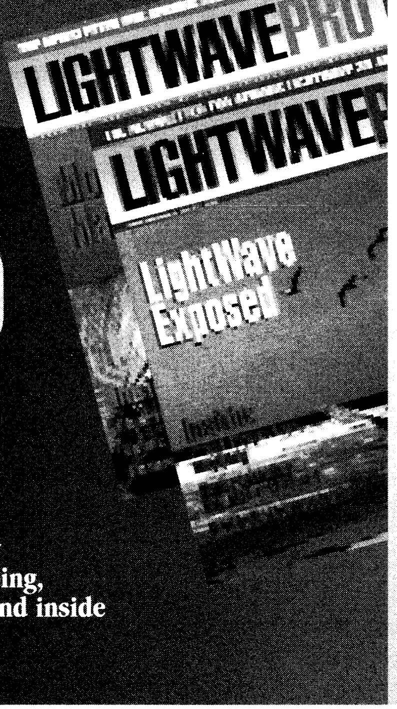
AVID PUBLICATIONS PRESENTS:

LIGHTWAVEPRO

The newsletter for serious LightWave 3D animators.

Learn the tricks of the trade from the industry professionals.

LIGHTWAVEPRO is loaded with in-depth articles by industry experts on modeling objects, texture mapping, image mapping, object deformation and product reviews, along with news and inside information about LightWave 3D.



Subscribe to LIGHTWAVEPRO today!

#1 Sign me up for a one year subscription of LIGHTWAVEPRO (12 monthly issues) for $72.00.

#2 SPECIAL - I currently subscribe to Video Toaster User. Please sign me up at the Special Subscribers rate of only $48.00 for 12 issues of LIGHTWAVEPRO.

#3 I want to subscribe to both LIGHTWAVEPRO and Video Toaster User at the Special Package rate of only $84.00.

(International rates are available)

Call Today to order with your credit card:

1-800-322-AVID

or send your check to:

AVID Publications
"LIGHTWAVEPRO"
273 N. Mathilda Ave.
Sunnyvale, CA 94086

Catch Up on Your LightWave Skills

SEND FOR BACK ISSUES OF LIGHTWAVEPRO

June 1994

LightWave Exposed; clip morphing; compositing with video; Grant Boucher shows how to bring your work into focus with the latest release of LightWave and several high-end rendering machines.

May 1994

Modeler Mania strikes *LWPRO*; Grant Boucher on recycling objects; understanding Modeler macros; spline patching pitfalls; introduction of Reader Speak column.

April 1994

Third-party products issue; complete reviews of LightWave products such as Humanoid, Pegger, Personal Animation Recorder, VertiSketch, WaveMaker, Sparks and Power Macros.

March 1994

Lighting issue; a beginner's guide to lens flares; real world lighting; Greg Teegarden shows how to create the effect of light bouncing off surfaces.

February 1994

Surfaces issue; introduction to surfacing; use LightWave's Alpha Channel to create eye-catching surfaces.

January 1994

Animation issue; Mark Thompson shows how the computer creates frames between points in time; compositing with live video; advanced motion techniques.

December 1993

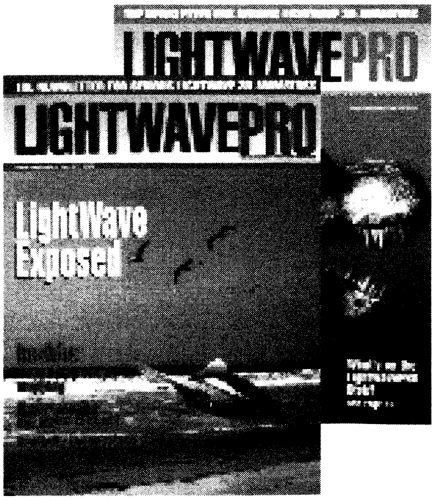
The author of Modeler 3.0/4000's macros, Arnie Cachelin, offers a how-to overview; a tutorial on making complex shapes with splines; John Gross shares his best Modeler secrets.

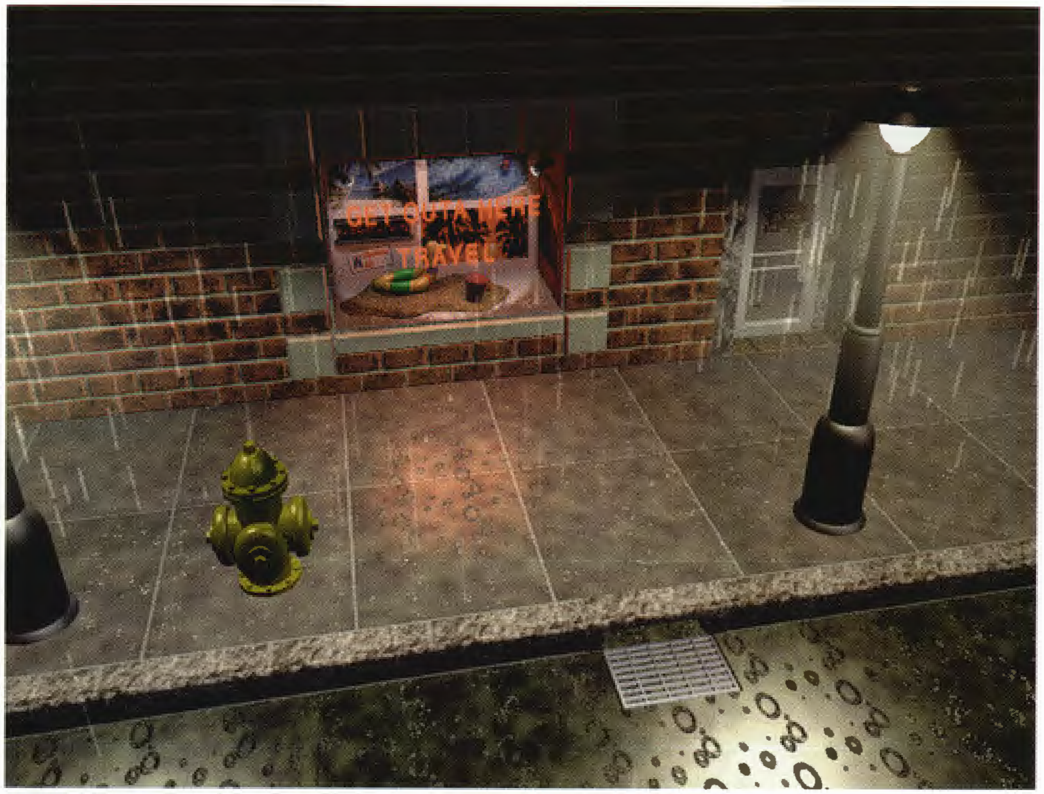
October 1993

LIGHTWAVEPRO's premiere issue; Grant Boucher's tutorial on displacement mapping; 24 Tips for Working with Bones; Breathing Life into LightWave

Each back issue is $10.00. Minimum credit card order is $20.00.

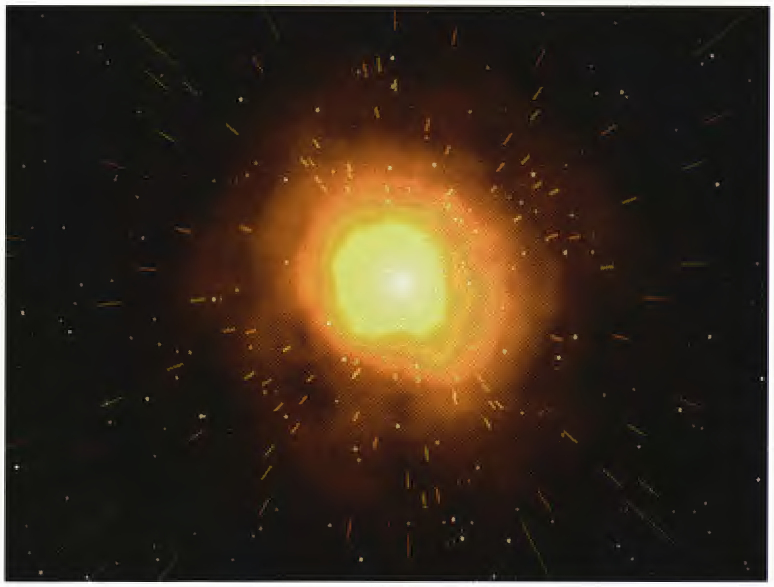
To place an order, please call 800-322-AVID.
Or write to:
AVID Publications
ATTN: Back Issues
273 N. Mathilda Ave.
Sunnyvale, CA 94086





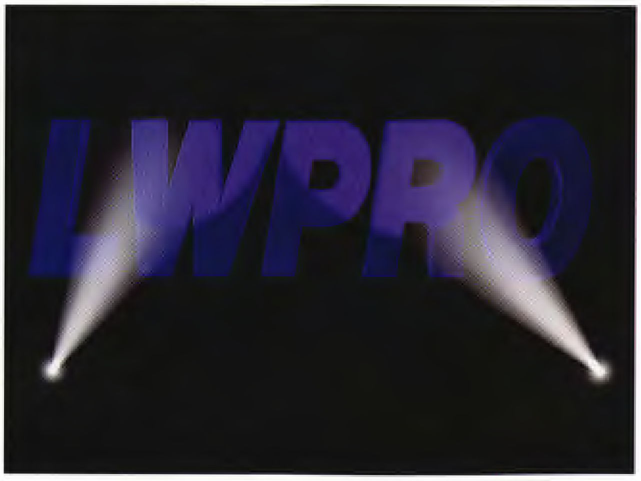
Rainy Night

Single point polygons with Particle Blur applied help create the illusion of a rainy night.
Copyright Mark Thompson 1992



Explosion

An animated Fractal Noise texture applied to morphed cone helps to create the effect of an explosion.
Copyright Mojo 1994



Light Beams

This simple image shows two cones with a light beam falloff image used in Transparency to create the effect of actual light beams.
Copyright John Gross 1994

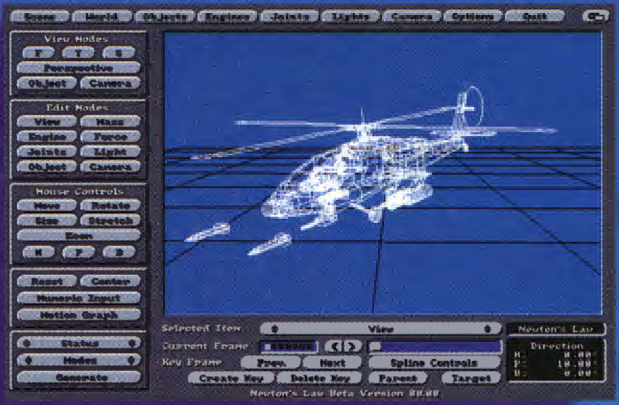
Newton's Law

Physics Module

Features:

- Mass
- Velocity
- Acceleration
- Gravity
- Magnetism
- Wind
- Friction
- Envelopes
- Joints
- Engines
- Springs
- Hinges
- Pistons
- Rotors
- Thrust
- Explosions
- Dampening
- Density
- Elasticity
- Rigidity
- Deformation
- Align to Path
- Materials Library
- Inverse Kinematics
- Accurate Collision Detection

- After Frames are generated by Newton's Law, choose to render/raytrace within your preferred graphics package, or on the WARP SYSTEM(TM) using Cybernetica's LIGHT ENGINE(TM).
- Use Newton's Law for the fastest and most accurate physics on the Amiga, and add Cybernetica's PHYSICS ENGINE(TM) on the WARP SYSTEM to get the fastest physics on any platform.



Scene File and Object Formats Supported:

LightWave
Wavefront
Imagine
and more...

Sir Issac Newton

Only $249

Call now for more information

1-(800) 697-8667

CYBERNETICA

Fax: (805) 730-7332

ART BY ARTASIA PRODUCTIONS/ IMAGES AND MOTION