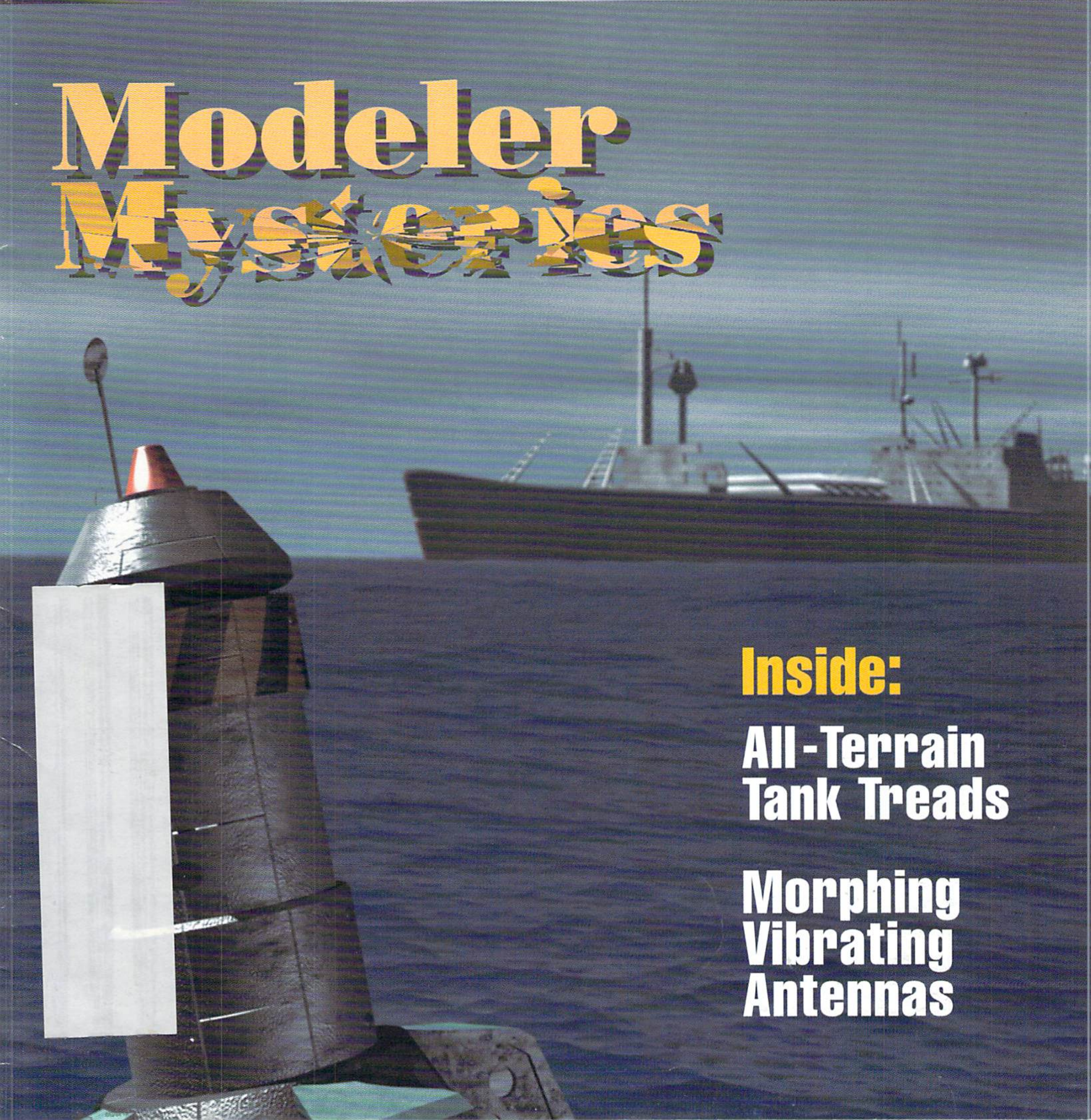
THE JOURNAL FOR LIGHTWAVE 3D® ANIMATORS

LIGHTWAVEPRO™

an Avid Media Group, Inc. newsletter

Volume 3 Issue Number 11 • November 1995 • $8.00

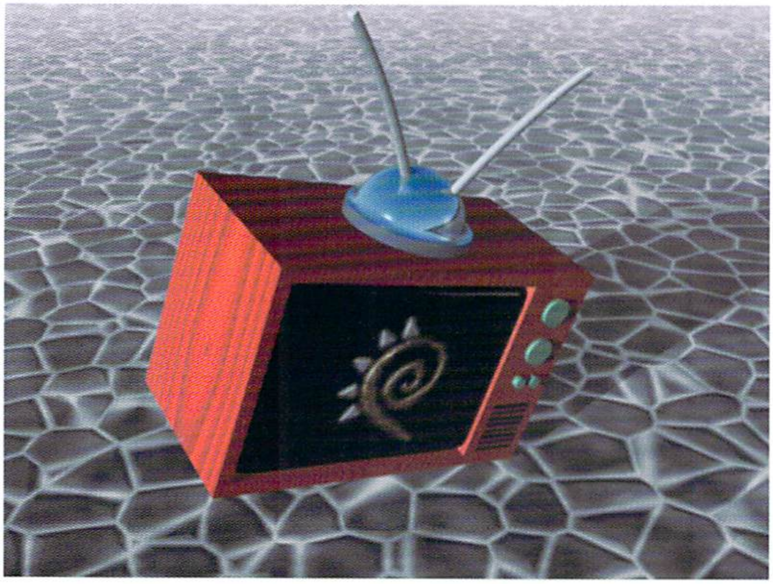
Modeler Mysteries



Inside:

All-Terrain Tank Treads

Morphing Vibrating Antennas

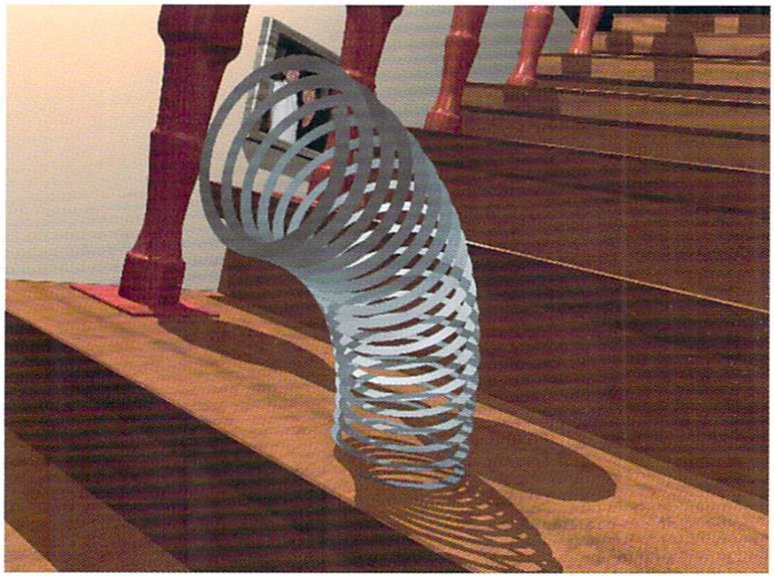


Antenna

A simple morph is used to create the quivering antennas on this TV. See "Mighty Morphin' Television Antennas," page 10.

Slinky

What walks downstairs alone or in pairs? With a simple Lathe operation and some bones, you've got a walking slinky. See "lwpro@internet.online," page 8 .



Tracks

For tank treads that look and act realistic, use a combination of a simple morph and strategic bone placements. See "Makin' Tracks," page 16 .

EDITOR'S MESSAGE

by John Gross

Well, another *Video Toaster User* Expo has come and gone and winter is in the air. The Expo seemed a bit smaller this year attendeewise, but there was a great deal of enthusiasm. Unfortunately, I didn't get to see all the booths I wanted, as time was limited and I ended up in a lot of conversations, but there were some impressive things. Not only were there very fast LightWave systems (read: Alpha), but the first of the commercial plug-ins were starting to appear. Among these were WaveFilter and Lock & Key (both for LightWave PC). You'll see some review/tutorials on these and other plug-ins in the future.

My vote for the coolest new thing at the show (as far as LightWave is concerned) is Fori Owurowa's Power View solid shader plug-in for Modeler. Using this product, you can bring up a window that will give you a solid, shaded preview of the object currently loaded in Modeler. If you have an OpenGL card in your PC, it will take advantage and give you much better feedback. The coolest thing of all about this plug-in is that NewTek is going to give it away for free on-line! (Way to go, NewTek!) For those of you who don't know Fori, he is a member of the LightWave programming team (along with Allen, Stuart and Arnie Cachelin) and the programmer of FreeForm, the Amiga-based spline modeling program.

Finally, I would feel neglectful if I didn't update you on my "PC situation." Last night, after about seven months, I finally put the cover back on my PC case. That's right—I now have a fully functional PC! I had to buy a new motherboard and replace the CPU and RAM from my old one, but it seems to be working pretty well. The story isn't over yet, though, as I am still taking my vendor to small claims court. *That* tale will have to wait for another day.

John Gross, editor

TABLE OF CONTENTS Vol. 3, No. 11

LIGHTWAVEPRO
an Avid Media Group, Inc. newsletter

"We provide the most valuable information to people who use technology to create messages with impact."

Editorial Office: Avid Media Group, Inc.
1308 Orleans Drive, Sunnyvale, CA 94089
Telephone: 408-743-9250; Fax: 408-743-9251
E-mail John Gross at jgross@netcom.com
Are you interested in writing for *LIGHTWAVEPRO* or submitting images? If so, contact us at our office or electronically.

LIGHTWAVEPRO (Vol. 3, No. 11); (ISSN 1076-7819) is published monthly by Avid Media Group, Inc., 1308 Orleans Drive, Sunnyvale, CA 94089. A one-year subscription (12 issues) in the U.S. and its possessions is $48 (U.S.); Canada/Mexico, $60 (U.S.); overseas, $84 (U.S.). To subscribe, call toll-free 1-800-322-2843. 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*, 1308 Orleans Drive, Sunnyvale, CA 94089.

The cover shot from seaQuest 2032 *shows a radio buoy, ejected from a Skipjack class submarine, surfacing off the bow of a Russian freighter in the year 1962. The scene is entirely LightWave-generated. The water surface consists of a flat plane mesh of about 100,000 polygons, with the crumple texture applied for the small waves and a physical displacement map for the larger waves. The water itself is black, no diffuse, with a reflectivity setting of about 60%. All modeling and animation by Greg Teegarden. Image copyright © 1995 Universal Television & Amblin Entertainment, courtesy of Amblin Imaging.*

Modeler Mysteries

Secrets of the Lost Tools

by Tom Williamson

Boy oh boy, those LightWave objects! We've all created plenty of scenes with them, but we're getting pretty tired of the living room scenes (complete with Beethoven, of course), and who *hasn't* used that damn Porsche? Most of you have probably moved on to object creation to bring your scenes to life. Some of you have even become modeling masters. The majority of animators just want to do animation, but they still need to get to know LightWave Modeler a little better. So let's get acquainted with some of the more enigmatic functions in our favorite modeling program.

Bend

O.K., I know this isn't a mysterious tool. But I seem to get a lot of questions on its use. Bend isn't very complex—it's just a little awkward at first. Knowing where to click when a bend is necessary seems to cause most of the confusion. I think of it this way: If I were going to reach into my computer screen and physically wrench on my model, I would grab the end I wanted, right where I wanted the bend, and pull one way or another.

Just pretend your model is stuck in space and that you're reaching in with a pair of pliers to grab the part you want to bend by left-clicking and dragging. Where you click has a major effect on how an object bends. This is difficult to explain and best left to experimentation. Remember, though, that the bend always occurs along the axis you are "sighting along." For instance, making a long, segmented tube along the Z-axis (which would appear as a disc in the Face view) and placing your cursor and bending from this view will result in your segmented tube bending as a pipe would along the Z-axis. Normally, you will be bending an object along its "length" axis, but certain interesting effects can be achieved by using Bend in the other axes.

If you need to bend the object in only one direction—with a semi-circular font logo, for example—hold down the Ctrl key (or the middle mouse button on a three-button mouse). This will limit the bend to the first axis you start moving in and bend angles of 15 degrees.

You've probably already noticed that, by default, the opposite side of the object is stationary, which is normal (and necessary in most instances). If you think you're grabbing the model in the right place but it's bending on the opposite side, you need to adjust your "sense." Just hit the Numeric button or the (n) key and click the (+) or (-) button next to Sense. This button controls which end of the object gets manipulated. When it is set to (+), the most "positive" side of the object (based on the positive and negative sides of the axis of bend) is affected; when it is set to (-), the "negative" side is affected.

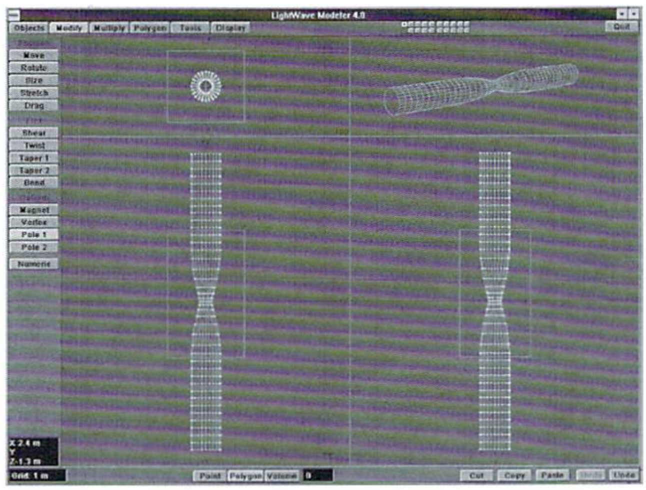


Figure 1: Using the Pole tool to create a venturi-like shape.

Magnet

Here's a wonderful feature that lets you push and pull on objects by using your pointer as a "magnet." To use Magnet, simply drag out a bounding box with the left mouse button (to determine the limits of the tool), and then use the right mouse button to enable the "magnet." The real trick is getting the box the right size and in the right position. When you start to drag the mouse, you'll notice the lines in the other views extend off the frame. This informs you that the area of influence extends forever along the axis you are sighting along. This can be adjusted (if you need it to be) by clicking in another view with the left mouse button and dragging the edges of the box where you need them.

You can also drag the whole box by left-clicking and dragging from the (+) in the center of the box. Remember that since the box represents a spherical area of influence, points near the outer edges (especially the corners) will not be affected. The (+) also serves an extremely important function—it determines where Magnet's greatest influence will be. Where you click when you start deforming the object also affects the way it works. Try to click close to the (+) for the most predictable results, but by all means experiment.

Pole

This is the tool I get the most questions about. It seems that the concept behind Pole is lost on most people, but in reality, it isn't all that complicated. Think of it as a Size tool with bounding box limits. The bounding box in this function works just like the one for magnetism (see above). After you've set up your bounding box, left-click and drag to scale the points within the box. As with Magnet's box, the greatest influence is at the center. (It falls off toward the outer edges.) With the exception of the bounding box, Pole works exactly like the Size tool. If you need to make a venturi-like shape in a pipe, for example, just build a cylinder with multiple segments, set up the bounding box where you need the constriction (with the (+) in the center of the cylinder), right-click and drag to the left. The middle of the pipe will shrink into a shapely hourglass figure (Figure 1)! Pole2 works in a similar way. This tool works like Stretch with a bounding box, allowing you to size differently on different axes.

Patch

Here's another cryptic tool, powerful but confusing. The basics behind it are that if you connect

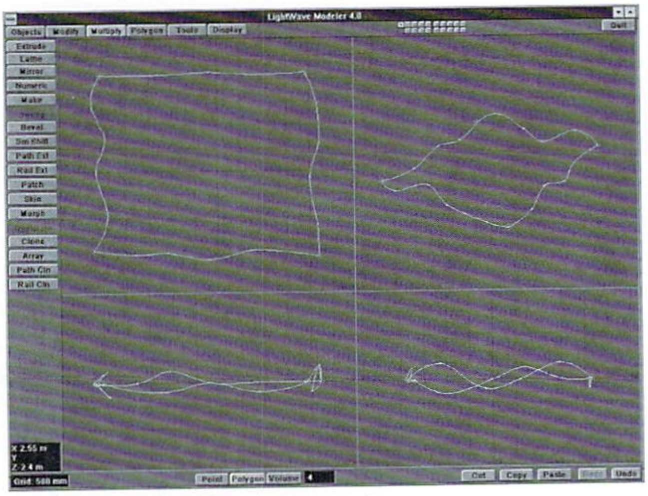


Figure 2: Four selected curves ready for patching.

enough curves together in just the right way, you can "patch" in between them. By patch, I mean you can create a polygon mesh that will approximate the curves to produce a solid form. To do this you have to have either three or four curves connected together (sharing points), which you can accomplish easily with the Weld tool. Since this isn't a tutorial I won't go into much detail, but let's make a Triscuit cracker from hell:

- Using Sketch, draw four curves (Figure 2) and connect them together by welding the points on the ends.

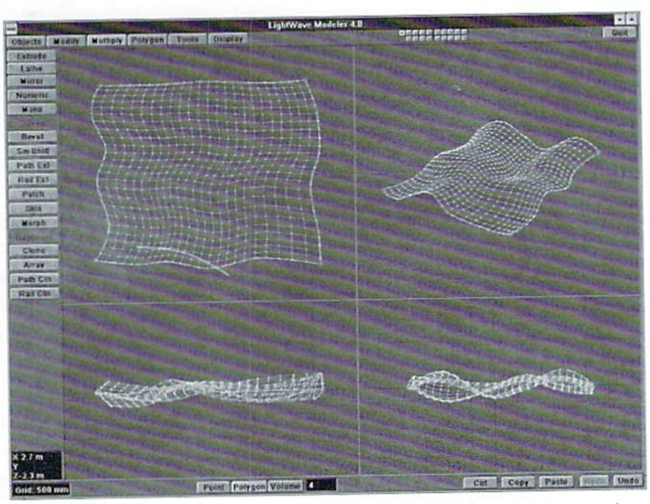


Figure 3: The four curves patched. (Make sure to check for non-planars when patching.)

- After selecting the curves one at a time, click the Patch (Shift-f) button. You'll be presented with a requester. The numeric fields are for the number of polygons you want your patch to consist of. The other buttons are for how you want the tool to approximate the curve: by Knots (more polygons at the bends) or by Length (equally spaced polygons). For now, use the default settings and hit OK.

The curves should now be patched. Depending on how crazy you went when you made your curves you should have a delicious looking 10 x 10 mesh cracker that pretty closely follows the shape of your curves (Figure 3). Mmm, yummy! Now try the tool again with an even smoother 20 x 20 mesh. With the Length buttons selected, you should get a more symmetrical mesh. You will most likely want to triple your polygons when you're finished, as this tool is notorious for generating non-planar polygons. You can use different Perpendicular and Parallel segments in your patches. These are always in relation to the last curve selected. Try changing the two values to see what happens.

Patch also works with three curves. The real trick here is the order in which you select your curves. The point that connects the first two curves you select becomes the radial point of the polygons. As with four-sided curves, the perpendicular and parallel segments are determined by the last curve selected. For three-sided curves, you should always patch "toward the corner." For more information on curves, see the "Curves" section.

Smooth Shift

This operation is very simple and very cool, but a lot of people have no clue as to what it does. Say you need a sphere with a raised ring around the center, like an orange wearing a belt (I don't know, it just popped into my head). All you need to do is select the polygons you want to be the "belt" and hit the Sm Shift (Ctrl–f) button. After entering the Offset (the thickness of the belt), hit OK, and there's your fashion plate Sunkist. What the tool does is take the polygons you have selected and move them the distance you specify along their normals (those little lines that tell you which way the polygon is facing). It also creates polygons to fill in the gap. A more practical example would be hair on a head. Just select the polygons you want to be the hair (after a Slice operation maybe, to create a separate surface?) and then offset the thickness you want the hair. I would recommend metaforming (under Subdivide) the resulting hair as well, just to make it a little nicer. Your humanoid never looked better, and he's not only the president, he's also a client.

Morph

When you have two polygons with the same point count, you can use Skin to put polygons between them. But what if you need intermediate sections? Morph to the rescue! This tool is also very simple: just select two polygons with the same point count, hit the Morph button, and set the number of segments. The result is a series of transitional segments that change from one shape to the next. I tried to come up with a real-world example, but I couldn't. In fact, I don't recall ever using this tool. But now you know what it does. *[Editor's note: I once made a chopstick using Morph. It allowed me to go from a rounded end to a squared-off end.—JG]*

Vortex

Vortex also works with your buddy, the bounding box. It acts like a tornado, twisting more at the center than at the edges (falling off completely at the edges of the bounding box, of course). Just left-click and drag out your box, and then right-click and drag to twist. The area where you click becomes the center of the vortex. This tool is great for creating swirling galaxies out of single-point polygons.

Array

Array does exactly what its name implies: it creates arrays of selected polygons. What's an array? Think of it as a grid with three dimensions. It works a bit like the Clone tool, allowing you to create multiple copies of your selected polygons, grouped together on as many axes as you specify. With one operation of this tool you can create a cube made up of tiny spheres that number 20 on the X-axis and 100 on the Y- and Z-axes. (It's possible to form an array with the Clone tool, but it would take multiple applications. Before the days of multiple Undo this was a pain.) Array also figures out your offset for you, spacing your multiplied polygons edge to edge. (The offset can be entered manually as well.) Keep in mind that the offset is calculated by the center of the polygons, not edge to edge.

Curves

Spline curves serve a number of purposes, and there are many ways to build them. The Sketch tool is a freehand curve creator. Just sketch out the shape you need and hit Make (Return). Your curve is approximated from your sketched line. You can also sketch with the right mouse button; when you let go, the curve is created automatically. If you need the ends connected, just drag points to the same place and use Weld (Ctrl–v). Once you get the curve in the right shape you can freeze it into a polygon. Just select it and hit the Freeze (Ctrl–d) button in the Tools menu. (Conversely, you could select the Numeric button (n) after drawing out your curve and select Polygon.) If you freeze an open curve, the ends will be connected for you.

The little "diamond" on one end of the curve is its start point. You can flip a curve like you flip a polygon,

See Modeler Mysteries, page 17

Cross-Platform Images

The Heartbreak of Non-Square Pixels

by Ernie Wright

When moving images across computer platforms or rendering for media other than video, how do you determine the pixel dimensions, and avoid the squashing and stretching that can occur?

I once caught a friend who has a Master of Science in structural engineering holding a ruler against the glass face of his computer monitor to measure some circles his program had drawn. The problem was that his circles weren't circular, and he wanted to know how much to correct them. (Many of you will remember the first time you drew a "circle" in DeluxePaint on the Amiga.)

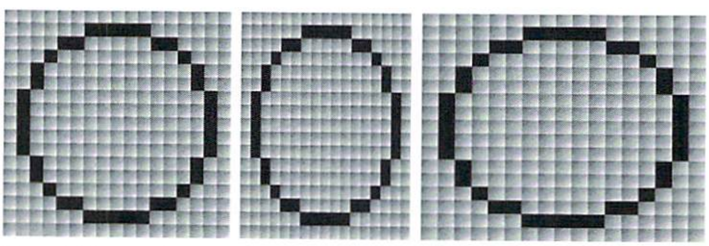


Figure 1: The same image displayed with different pixel aspect ratios.

Being an engineer, and therefore an empiricist by definition, my friend regarded his measurements rather seriously, as if they had a reality that was independent of the hardware and the methods he happened to be using. But his program was going to be used by other people on other machines. What if he'd used a different size monitor or a different screen mode, or a tape measure that conformed to the curvature of the screen? What if he'd measured near the edge of the screen rather than the center, or early in the morning, before the components of his monitor had reached thermal equilibrium? What if a friend of his was about to sneak up behind him and mess up his monitor adjustments?

Unless you've done some graphics programming, the arcana of pixel and image aspect ratios and the transformations required to map an abstract image to a particular output device are probably unfamiliar. Even if you're a graphics programmer (or an engineer), you most likely learned about non-square pixels on the street, because no textbook author wants to acknowledge their existence, much less try to explain them. (The references I normally reach for first—Foley, van Dam et al., Watt, the Graphics Gems books—have nothing to say on this subject, while the lower-tier, $30 paperbacks get almost everything wrong.)

This article won't entirely fill the void, but I'll try to give you enough background and practical advice to make you comfortable with the subject and ready to ask the right questions.

Aspect Ratio

Aspect ratio is the ratio of a rectangle's width to its height. Maybe you've always heard this term applied specifically to images, pixels and screens, but it's meaningful for any rectangle. It was used 90 years ago to refer to the shape of an airplane's wings (and is still used for that purpose, although the definition has been modified to account for the fact that, by and large, airplanes no longer have rectangular wings). The aspect ratio of U.S. letter-size paper is 8.5/11 = 0.773; for many front doors, it's 36/80 = 0.45; and for the shelves in my Swedish bookcases, it's 65/26 = 2.5.

You'll often see an aspect ratio written only as an explicit ratio of two numbers—"4/3," for example, or even "4:3," with a colon in the middle. Don't be misled by this. Aspect ratio is a single value, the result of dividing the first number by the second one. Every square has an aspect ratio of 1.0. Conversely, every rectangle with the same aspect ratio has the same shape.

We'll be dealing with two different aspect ratios in the discussion that follows. Image (or frame) aspect is the shape of the entire image. Pixel aspect is the shape of each pixel in an image. It's important not to confuse the two.

A Frequently Asked Question

"How do I convert a 752x480 LightWave image so that it displays correctly on a PC (or on almost any computer display other than the Amiga's)?"

Most computer displays use square pixels while NTSC/D2 does not, so you need to correct for the pixel aspect. You have a couple of options.

If the frames haven't been rendered yet, you can use LightWave to set the Pixel Aspect Ratio (on the Camera panel) to Square Pixels. By default, this option sets the resolution to 640x480, which gives you an image aspect ratio of 4/3 = 1.333, very nearly the same as the 1.346 image aspect for 752x480 with NTSC/D2 pixels. (In versions of LightWave prior to 4.0, the pixel aspect setting is ignored when a Custom Size is being used. I'll have more to say about this a little later.)

If you use LightWave 4.0 primarily to generate images for display on PCs, you can edit the LightWave configuration file so that Square Pixels is your default pixel aspect. The LightWave configuration file is called LW-config on the Amiga, LW.CFG on platforms running Microsoft Windows and .lwrc on SGI machines. Find the line containing "DefaultPixelAspect" and change the value to 2. (The value is the position of the option in the Pixel Aspect Ratio pop-up list, counting from 0.) If you're using an earlier version of LightWave, you might want to create a scene file named "default" (or whatever you like) in which to store default settings that, like this one, can't be added to the config file.

If the frames have already been rendered, you'll need to scale them in an image-processing program. You can either reduce the width to 640 or increase the height to 564. Reducing the width is preferred because it's less likely to introduce scaling artifacts—from an image-quality point of view, it's better to slightly reduce the amount of information in an image than it is to try to add information that doesn't really exist.

It usually isn't necessary, but if you want, you can account for the slight difference in image aspect, either by cropping before scaling or by scaling to slightly different dimensions. The crop should remove seven pixel columns, divided any way you like between the left and right sides of the image, so that the new width is 745. You can also scale to either 646x480 or 752x559, though you may find that these "unusual"

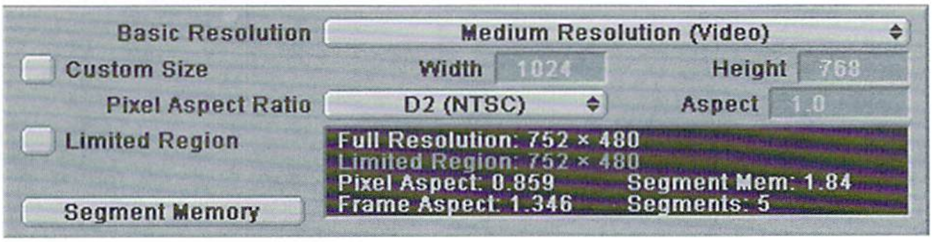


Figure 2: The Camera panel's resolution controls.

sizes aren't as convenient to use in other applications.

What about PAL/D2?

Pretty much the same deal. You won't go terribly wrong by scaling to 640x480, but to preserve more of the image information, you can also scale to 752x565 (height reduction) or 766x576 (width enlargement). Or you can decide not to mess with the images at all. LightWave gives PAL/D2 pixels an aspect of 1.019, which is well within the range of variation for most devices claiming to use square pixels.

Magic Formulas

At this point, you might be wondering where all of these numbers are coming from. Well, there are formulas. They can be a little confusing—there are three aspect ratios and four pixel dimensions involved—so don't expect all of this to be instantly clear. In practice, you'll probably find yourself using just one or two of the formulas all the time, and if you need them often enough, the calculations will become second nature.

Let's define some shorthand references to the values:

$w1$, $h1$—the width and height of the source image

$w2$, $h2$ — the width and height of the target image

$p1$ — the pixel aspect of the source image

$p2$—the pixel aspect of the target device

i— the image aspect (the shape of the image)

The aspect ratio of an image can be found with either of the following:

$i = (w1 * p1)/h1$

$i = w1/(h1/p1)$

When the pixels are square, $p1 = 1.0$ and both formulas reduce to the definition of aspect ratio, $i = w/h$.

To correct an image for distortion-free output on a different device (no change in image aspect), use one of the following:

$w2 = p1 * w1/p2$

$h2 = p2 * h1/p1$

Don't use both! You'll be applying the same correction twice. I recommend using the formula that results in a reduction of one of the pixel dimensions whenever possible. If $p1 < p2$, then use the first one (reduce the width). If $p1 > p2$, use the second one (reduce the height). One of the pixel aspects (either $p1$ or $p2$) will often be 1.0, which can simplify the calculation. (And if $p1 = p2$, the formulas collapse to $w2 = w1$ and $h2 = h1$, of course.)

When you know both the image aspect and pixel aspect and you have one of the pixel dimensions in mind, you can find the other one with this formula:

$w1 = h1 * (i/p1)$

$h1 = w1/(i/p1)$

You have to know one of the two pixel dimensions before you can calculate the other one; this is how you determine the scale of the image.

Finally, to calculate a pixel aspect for an image with a known image aspect and dimensions, use the following:

$p1 = (h1 * i) / w1$

For displays that use square pixels, $w1 = h1 * i$ and $p1 = 1.0$.

Custom Size

If you'd like to apply these formulas to custom-size images and you're using LightWave 4.0, hakuna matata. Turn on the Custom Size checkbox, select Custom from the Pixel Aspect Ratio pop-up list, and enter values for Width, Height and (pixel) Aspect. LightWave displays the frame (image) aspect, i, in the information box next to the resolution controls. Typically, the pixel aspect will be known, and you'll have some idea of what the image aspect and one of the two dimensions should be. You'd use these in the formula for w1 or h1.

If you're using Custom Size in an earlier version of LightWave, you'll have to do the pixel aspect correction outside LightWave, in an image-processing program. When rendering custom-size images, versions 3.5 and earlier of LightWave ignore the Pixel Aspect Ratio setting and instead always use NTSC/D2 pixels, with an aspect of 0.859. (This is a holdover from LightWave's origin as an application for the Video Toaster. The resolution and pixel aspect controls in LightWave 3.5, the first version not tied directly to the Toaster, hadn't yet made a complete transition to device-independent rendering.) The approach when using pre-4.0 LightWave is to set the Width and Height so that the image aspect is correct for pixels with NTSC/D2 aspect, then scale the rendered image to account for the intended pixel aspect.

Suppose you want to use LightWave 3.5 to render a 300x300 image for display on a device that uses square pixels. Since the need for Custom Size with square pixels is common, it's worth stating a rule of thumb here: Divide the desired width by 0.859, render, then scale to the desired width. So for a 300x300 square-pixel image, set the Width to 300/0.859 = 349 and the Height to 300, render, then scale the rendered image back to 300x300.

In general, you first need to convert from the desired pixel dimensions to the NTSC/D2 dimensions. Use the formula for w2 or h2, whichever makes the pixel dimension larger. Set w1, h1 and p1 to the desired width, height and pixel aspect, and set p2 = 0.859. Render at the new pixel dimensions (w2xh1 or w1xh2), then reduce the rendered image to w1xh1.

Hopefully I've shed some light on the oft-confusing aspect ratio question. Learning these values and applying them when needed will give you the image you're looking for!

LWP

For the past five years, Ernie Wright has been a programmer doing scientific visualization for several defense-related agencies. His code can also be found in LightWave (the HAM routines in 3.5 and wireframe previews in 4.0 for Windows). Ernie works from his home in Maryland while watching his 2-year-old daughter, whose little brother is due in January 1996.

Statement of Ownership

Post Office Notice: Statement of Ownership, Management and Circulation for second-class privleges as required by 39 USC3685: 1. Title of Publication: LIGHTWAVEPRO 2. Publication No. 10767819 3. Date of Filing: September 29, 1995 4. Frequency of Issue: Monthly 5. No. of issues published annually: 12. 6. Annual subscription price: $48 7. Complete mailing address of known office of publication: 1308 Orleans Drive, Sunnyvale, CA 94086 8. Complete mailing address of the headquarters of general business offices of publishers: Michael D. Kornet, 1308 Orleans Drive, Sunnyvale, CA 94086 9. Full names and complete mailing addresses of publisher, editor and managing editor: Publisher: Michael D. Kornet, 1308 Orleans Drive, Sunnyvale, CA 94089, Editor: John Gross, 1308 Orleans Drive, Sunnyvale, CA 94089, Managing Editor: Corey Cohen, 1308 Orleans Drive, Sunnyvale, CA 94089 10. Avid Media Group, Inc., 1308 Orleans Drive, Sunnyvale, CA 94089, James C. Plant, 1308 Orleans Drive, Sunnyvale, CA 94089, Michael D. Kornet, 1308 Orleans Drive, Sunnyvale, CA 94086 11. N/A 12. N/A 13. Publication Name: LIGHTWAVEPRO 14.Issue date for circulation data below: September 15. Extent and nature of circulation given in this order, number of average copies each issue during preceding 12 months followed by actual number of issues published nearest filing date, A. Total no. of copies 5,561 & 4,300 B. Paid and/or requested circulation 1. Sales through dealers and carriers, street vendors and counter sales: 446 & 250 2. Paid or requested mail subscriptions: 2,365 & 2,802 C.Total paid and/or requested circulation: 2,811 & 3,052 D. Free distribution by mail, carrier or other means, samples, complimentary and other free copies: 11 & 20 E. Free distribution outside the mail: 1,055 & 240 F. Total free distribution: 1,066 & 262 G. Total distribution: 3,877 & 3,314 H. Copies not distributed 1. Office use, left over, unaccounted, spoiled after printing: 1,684 & 986 2. Returns from News Agents: 0 & 0 I. Total: 5,561 & 4,300 16: This statement of ownership will be printed in the October issue of this publication. 17: I certify that the statements made by me above are correct and complete. Michael D. Kornet, Publisher.

lwpro@internet.online

by Dan Ablan

Welcome to another installment of "lwpro@internet.online." The World Wide Web is growing fast, and when it comes to LightWave, there have been many advances as well. If you subscribe, like I do, to the LightWave mailing list (see sidebar for information), you may have noticed some really frantic activity over the past month. Thankfully, most people have broken away from the whining and whimpering about whose system is bigger, better and faster; now they're concentrating on getting involved in some cool animations. With the final 4.0 release hitting the streets, there are fewer bugs and problems, which means all we have left to do is use the programs! I have my final 4.0 and am pretty pleased with it. Every once in a while I think back to the 1.0 and 2.0 LightWave days. My, how far we've come.

Anyway, let's get to what's been happening on the Net. On the mailing list, there're all kinds of topics to choose from. People have been posting questions about field rendering, glow features, laser beams, pools of water and all kinds of stuff to choose from. On the newsgroup list, comp.graphics.apps.lightwave, many of these same topics have been popping up, as well as many others. Topics of discussion include women in animation (Mom would be proud to know), cracks (never a good subject) and LightWave in commercial television (e.g., blue M&Ms and *Space: Above and Beyond*). *[Editor's note: For feature coverage of* Space *and the M&Ms—plus exclusive tutorials from the animators behind their LightWave effects—see the November 1995 and January 1996 issues of* Video Toaster User, *respectively.—JG]*

There's been chat about almost everything, so picking topics for this month's article was relatively easy.

You Slinky Thing

Oddly enough, I noticed a lot of talk about creating a slinky in LightWave. Some people responded with ideas, others with complete parameters for modeling the object. I've played around in Modeler a bit, and this is what I came up with:

- Enter Modeler. Select Box (Objects), and in the numeric requester (n), enter these values:

	Low	High	Segments
X	176 mm	253 mm	1
Y	0	1 mm	1
Z	0	0	1

- Next, select Lathe (Multiply) and then Numeric (n). Enter these values first:

  Start 0
  End 6600
  Sides 800
  Offset 200 mm
  Axis Y
  Center 0, 0, 0

- This will produce a short slinky. This slinky object can be used for just, well, sitting around. Save it as "slinkyshort.lwo". slinkyshort.lwo will also be used to morph to the next slinky object, which should be created by applying these values to the original polygon:

  Start 0
  End 6600
  Sides 800
  Offset 1.5 m
  Axis Y
  Center 0, 0, 0

- These settings will result in a tall slinky that has been pulled apart. Save this object as "slinkytall.lwo".
- In Layout, load the short and tall slinky objects. In the Objects panel, select slinkyshort.lwo as the current object, then select slinkytall.lwo as the Metamorph Target.
- Click the (E) button next to Metamorph Amount. Create a keyframe at 5. Create another at frame 50. At frame 50, set the value to 100%, and select Spline Controls (s) and input a Tension of 1.0. At frame 5, set spline tension to 1.0 also. Click OK.
- Select slinkytall.lwo as the current object. Bring Object Dissolve to 100%.
- Finally, set the Last Frame (Scene panel) to 50 and save the scene.
- Make a preview, and watch the slinky grow!

We morphed this object because it will look more realistic in an animation. You could have easily loaded just the slinkytall.lwo object, and stretched it on the Y-axis over time, but the result would look like it was stretched. The morph, on the other hand, gives the appearance of springlike movement. And speaking of springs, try varying the above settings for Box and Lathe to create a nice-looking spring or coil.

But we all know that slinkies walk down stairs, and it's a wonderful thing, right? It's fun for a girl and a boy, so read on.

I experimented with a series of morphs—from short slinky to bent slinky to tall slinky—and it just wasn't looking right. So, along came bones. The image in the color pages shows the final bent slinky a la bones. Figure 1 shows the bone structure I set up for a wonderful toy. I set up four base bones to anchor the slinky when it bends. I configured a hierarchy such that if I decided to move the base to make the slinky jump and come to life, the other bones would follow. Now let's see who can make it walk down all those stairs.

Other Interesting Topics

There have been a number of posts on the LightWave mailing list about 4.0's new **Glow** feature and field rendering. By now, if you have been keeping tabs on the mailing list, you'll see that the Glow option seems to flicker when used in conjunction with field rendering. I'll find out more about this and include the information in an upcoming article. And, with that, since it will be more of a technical column, I'll include information about color matching and LightWave. It seems a whole discussion has started regarding LightWave, the PAR, the Toaster and NTSC video signals.

A few people on the LightWave newsgroup were discussing the proper way to make a snake slither. After your snake is modeled, set up and size one bone. From there, use the Add Child Bone (=) feature. Turn off the Move functions for X, Y and Z, then turn off the Pitch and Bank for the rotation.

Move the bones back and forth from the head on back. There you go—a slithering snake. However, this procedure would take a long time, so taking it a step further, add the bones in an Inverse Kinematics chain and just move the null. Check out *LIGHTWAVEPRO* author Mark Thompson's LightWave in.focus, Volume 1 training tape for a great tutorial along these lines.

Another person asked how to make random stars. If you are using Amiga LightWave, you'll have the Points to Polygon macro. If you are using PC LightWave, use the Points to Polygon function located in the Custom pop-up in the Tools menu (make sure you add all of the Modeler plug-ins first). In either case, make random points in Modeler on a very large scale and use the Points to Polygons function to create single-point polygons (particles). For added measure, make two sets offset from each other. Under the objects panel in Layout, select one set of stars with Small Particle/line Size. Set the other to Medium. This will add variation to your star sizes when rendered. Place a fractal noise reflection map on one of the star objects to give the stars a shimmer.

Bookmarks

Here's a list of some interesting LightWave-related Web sites for you to check out:

http://www.cabot.nf.ca:80/~sean/
http://www.creativedir.com/
http://www.iomega.com/
http://www.lei.net:8080/dml/digitalsite2/digitalsite.html
http://www.mt-inc.com/MT.html
http://www.ci.chi.il.us/
http://www.lightside.com/~dani/
http://www.winternet.com/~derekb/
http://www.research.digital.com/SRC/3D-animate/
http://info.acm.org/~joyces/homepage.html

If you or anyone you know has a cool LightWave-related Web site, please contact me with the information so I can include it in an upcoming article.

Remember to check out the LightWave newsgroup and mailing list when you can, and don't be afraid to post a query, even if you feel it's a dumb one. No question is dumb if it's going to help you understand how to become a better animator.

LWP

Dan Ablan is the president of AGA Digital Studios, a LightWave-based animation company in Chicago, Ill. Dan can be reached at dma@mcs.com, or the old way at (312) 649-9391. Check out AGA's WWW site at http://www.mcs.com/~dma/home.html.

Subscribing to the LightWave Mailing List

To subscribe to the LightWave mailing list, simply send e-mail to lightwaverequest@webom.com and type "Subscribe" in the body of your message. If you wish the mail to go to an address other than the one you are posting from, include the full address after the Subscribe statement. After a day or so, you will get a message confirming that you have joined and informing you that you will start receiving mail sent to the list.

Sometimes the amount of mail can be overwhelming, and if you can't check it frequently enough, you may want to subscribe to the digest version. With this version, all the messages for a week or so are packed together as one file. To subscribe this way, send a Subscribe request as outlined above to lightwavedigestrequest@webcom.com.

—John Gross

Mighty Morphin' Television Antennas

by Alan Chan

The scene opens to a huge warehouse at night. The stiletto chirpings of crickets overwhelm the senses as CGIMan steps into the warehouse, hot on the pursuit of the field-rendered Video Wizard.

As CGIMan looks up, he sees a television set heading straight for his head! With speed-of-light reaction, he sidesteps the box of electronic doom, and it bounces off the floor harmlessly, antennas vibrating from the landing...

CUT TO REALITY

Whether you're animating the next CGI epic or just simple flying logos, there are times when you're faced with the challenges of a flexing object. Car and television antennas and the like vibrate when hit, slowly coming back to a standstill over a good number of cycles. So how can we simulate this phenomenon in LightWave? There is a quicker way than setting keyframes until your eyes hurt: morph targets.

To explain the idea, let's use the plot device in our most excellent script opener. We'll animate a falling television set.

The Theory

We will use multiple morph targets to realize this vibrating antenna (animating CGIMan is up to you). To this end, let's brush up on our morphing theories.

(1) At the absolute base level of a morph, LightWave takes a certain point from the source object and the same point (determined by the point order) from the target object, and calculates the morphed point based on the Metamorph Amount at that particular frame. For instance, if the first point in the source object is at 0, 0, 0 and the first point of the target object is at 6, 0, 0, a Morph Value of 50% will cause the morphed point to be at 3, 0, 0 (Figure 1). Repeat this procedure for the number of points you have in your object, and you will end up with a 50 percent morphed object. So you can probably see why objects with the wrong point order turn into a mess of points instead of materializing into the target object, and why you need to have objects with identical point counts.

(2) The source object is the one that is affected, not the target object. If you set a Metamorph Target and a Metamorph Amount for an object, you will see the changes in that object, not in the Morph Target. The target object serves only as a set of data that LightWave uses to reshape the source object.

(3) To use the Metamorph Target object as a morph target, LightWave requires that you load it into the scene file. Therefore, the standard operating procedure is to hide it so that it does not appear in the final rendered sequence as a separate object. That's why you would usually set the target object's Object Dissolve envelope

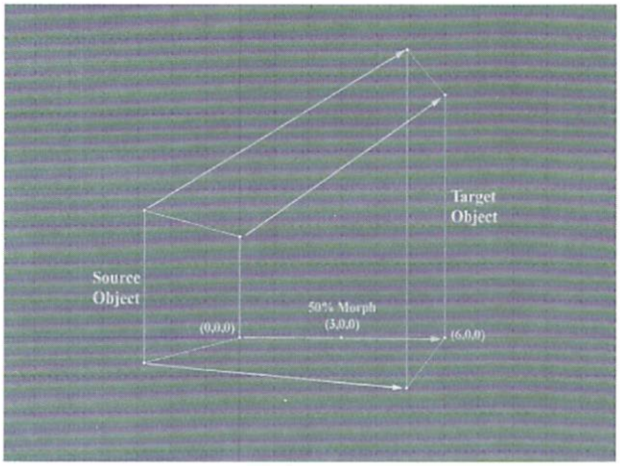


Figure 1: LightWave's linear morph.

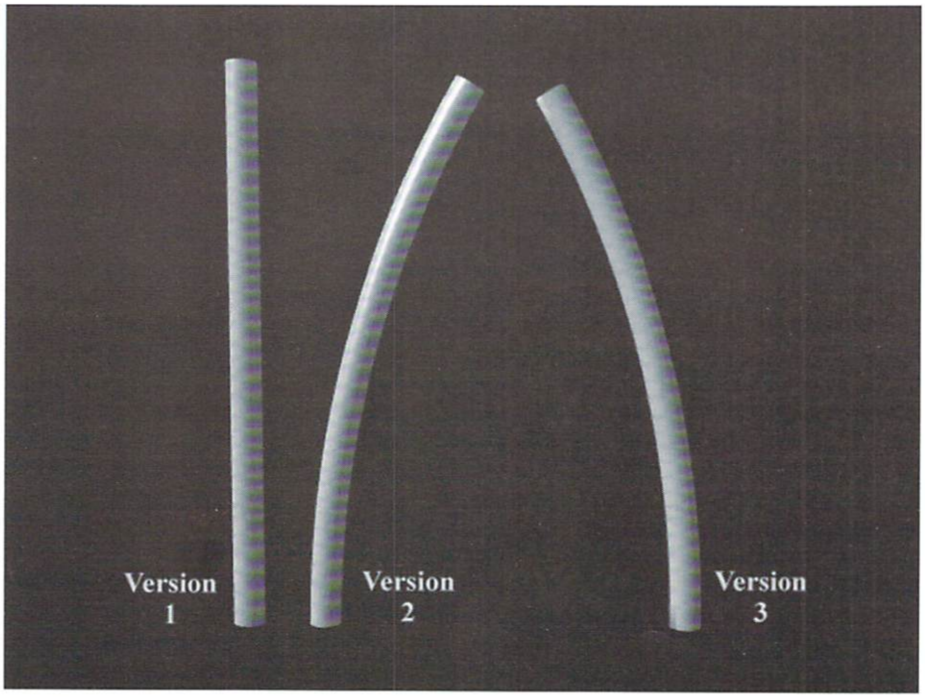


Figure 2: Objects we will use to simulate a vibrating antenna.

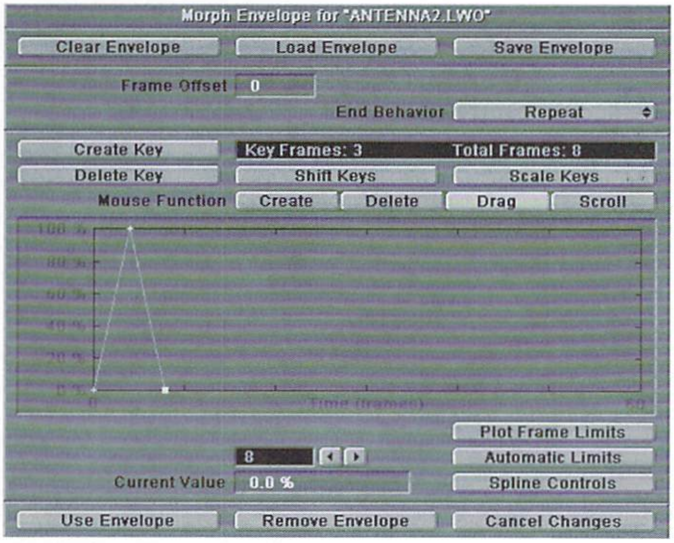


Figure 3: An envelope to generate a repeating vibration in the antenna.

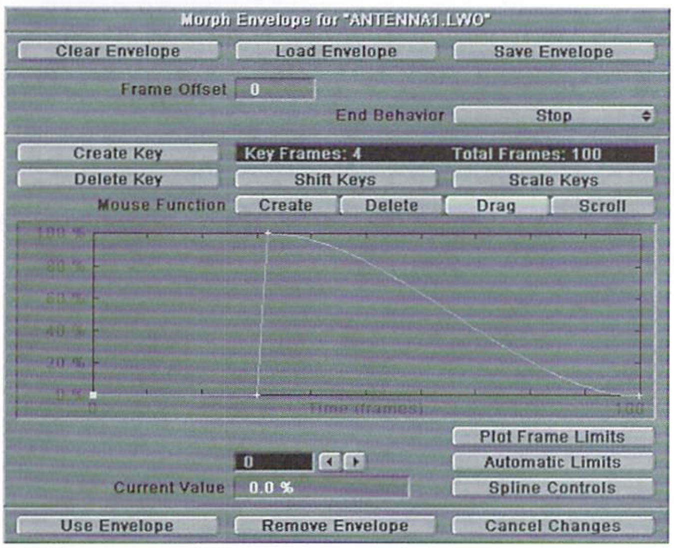


Figure 4: An envelope to control the amount of "bounce" in the antenna.

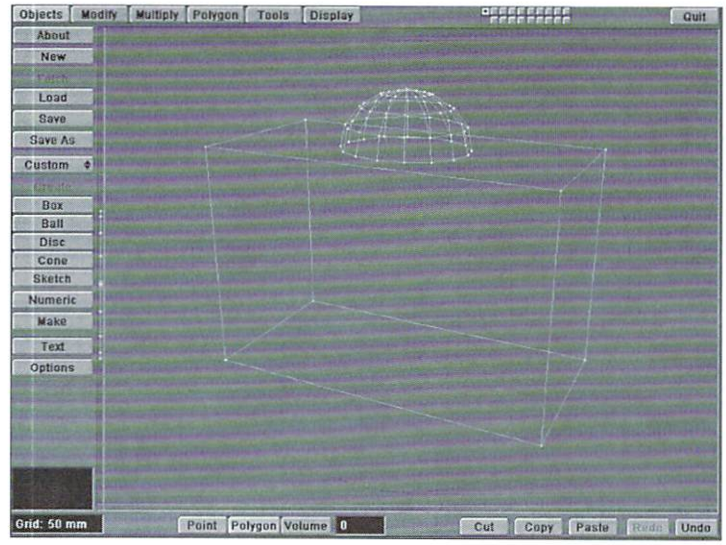


Figure 5: A wonderfully simple TV set object.

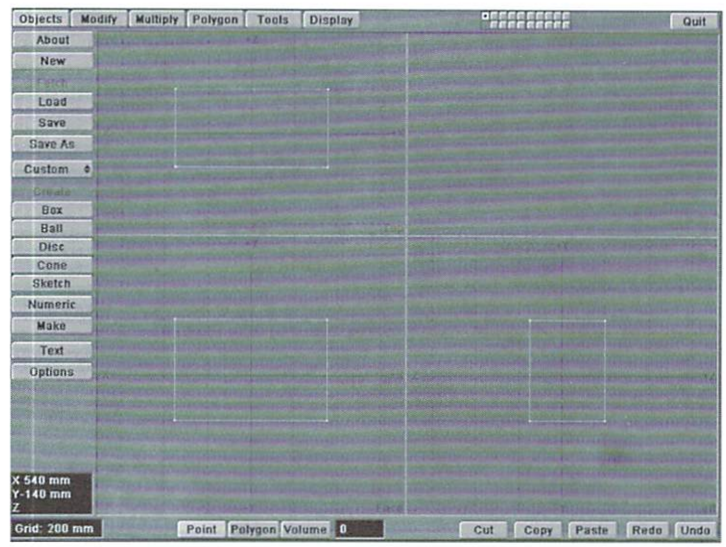


Figure 6: Dragging out a box shape for the TV set.

to 100%, or move the object out of camera view. (In addition to setting the Object Dissolve to 100%, I also parent my target object to the source object. Though it's a somewhat unnecessary step, in bounding box form, it allows me to use the dotted line bounding box to estimate the morphed position.)

(4) We can specify multiple morph targets for an object in the following way: After specifying a target object for the source object, assign a Metamorph Target to the target object itself. After the source object is morphed into the target object, the target object itself can then be morphed into a third object (Figure 2). With this "layered" morphing method, we can perform one morph after another to the same source object.

Irrelevant to this tutorial, but worthy of note: With the Morph Surfaces option, LightWave also allows for morphing surface values, but surface morphing can only be achieved from the source object to the first target object. There are no provisions for layered surface morphs.

The Execution

Having refreshed our thoughts on morph theories, it's time to take these concepts and put them to work.

Let's say we have three versions of an antenna (Figure 2). The first version represents the antenna at rest, or as it appears before anything happens to it. The second version is bent to the right, while the third version is bent to the left, representing the extreme angles to which the antenna will vibrate. All three versions have the same point count and polygon order; in short, they are morphable.

If we use a repeating envelope (Figure 3) to set version 2 so it's morphing into version 3, we will get a vibrating antenna, right? Easy enough. But the antenna doesn't stop vibrating. In reality, the device would begin vibrating violently when struck (as the television set hits the ground), but the vibration would slow down and cease over time. So how do we achieve this? The answer lies in a layered morph.

If we use version 1 as our antenna (the one that's actually attached to the television set) and create a morph envelope (Figure 4), we can morph version 1, a static antenna, into version 2, a vibrating antenna (which is repeatedly morphing between the two bent antennas). When we envelope the morph from 0% to 100% in one or two frames, we'll make the antenna start vibrating as if a force has been applied to it. As we start decreasing the Metamorph Amount over time, the vibrations become less and less noticeable,

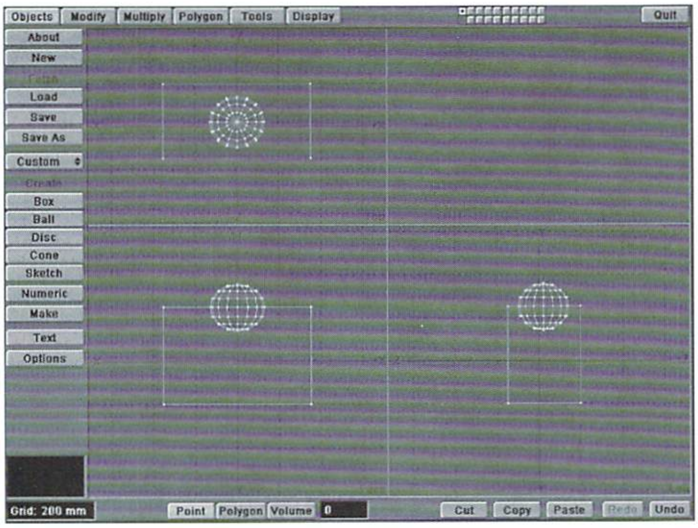


Figure 7: Dragging out a ball shape.

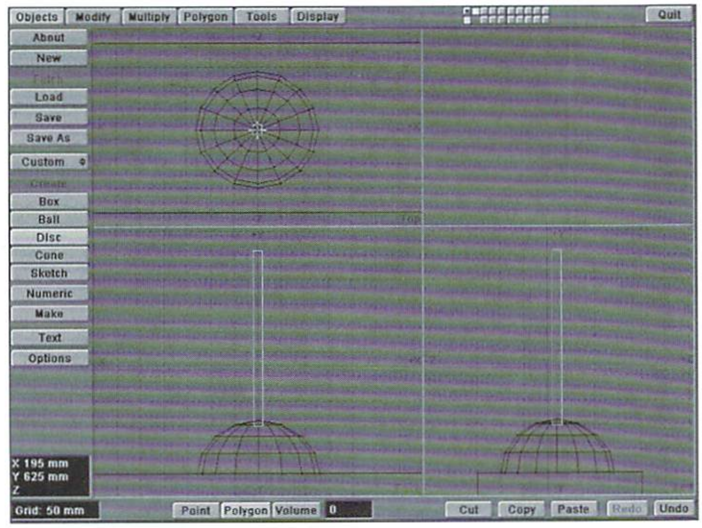


Figure 9: Modeling the antenna.

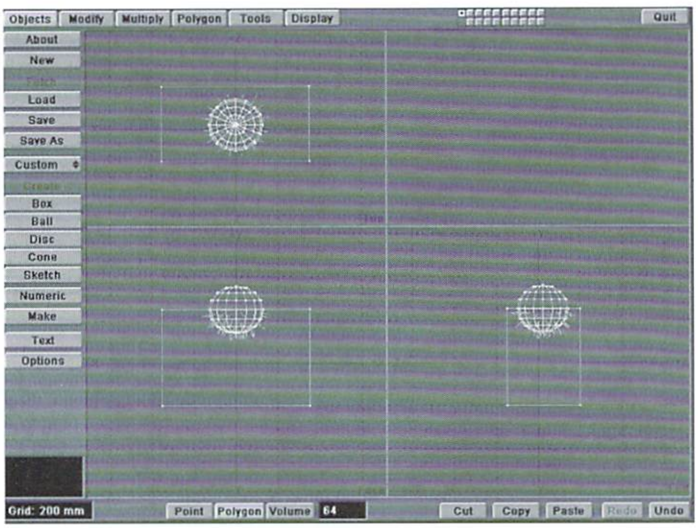


Figure 8: Trimming the ball into an antenna base.

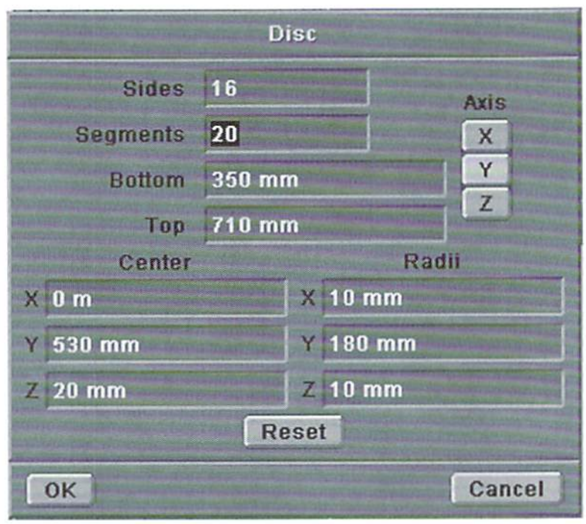


Figure 10: Numeric Options panel for Disc.

as if they are being reduced by physical forces such as air friction. Finally, when the Metamorph Amount has returned to 0%, the antenna has returned to its first, static position.

ACT I

- In Modeler, let's build a television object with antennas on top of it. We'll model an extremely simple TV such as the one in Figure 5. First, drag out a box for the television proper, and Make it (Figure 6). Next, drag out a ball for the base of the antenna at the top of the television set (Figure 7). Select the polygons making up the bottom half of the antenna and cut it off (Figure 8). You now have a simple television set object. Make sure you name your surfaces accordingly and save your TV object before continuing. (You may, of course, want to model a more sophisticated TV set.)
- Go to an empty foreground layer and place the TV box in a background layer. Using the Disc tool (Objects menu) drag out a really thin, upright antenna (Figure 9). Click on Numeric (n) to bring up the Numeric Options panel and enter 20 or so for the Number of Segments value (Figure 10). Click OK, then hit Make (return). This will give you a long, segmented antenna object.
- Since we're going to be bending and twisting our antenna, we will want to make sure that the bending process will not create non-planar polygons. Tripling our polygons beforehand will do the trick. Making sure that you are in Polygon Select mode and have nothing selected, go to the Polygon menu and click Triple (Shift–t) to triple every polygon (Figure 11). Save this object as "Antenna1.lwo".
- Select the Bend tool (Modify menu). In the top view, place your cursor in the middle of the disc. While holding down the Ctrl key (to restrict movements to one axis), click and drag the mouse so that the top of the antenna bends slightly to one side. Thirty degrees is a fairly decent amount by which to bend the antenna (Figure 12). Save this object as "Antenna2.lwo".
- Returning to the Modify menu, select the Bend tool again, and click Undo to return to the original upright antenna. Then click Numeric (n). In Bend's numeric panel, find the Angle requester (Figure 13) and make its value negative (e.g., if the value is 30, make it -30, and if the value is -30, make it 30). Do not modify any of the other parameters; they have already been set up from the last Bend operation (which created the Antenna2.lwo object), and we want to make an antenna that is exactly identical except that it bends the opposite way. Click OK to use these values, and save the object as "Antenna3.lwo".

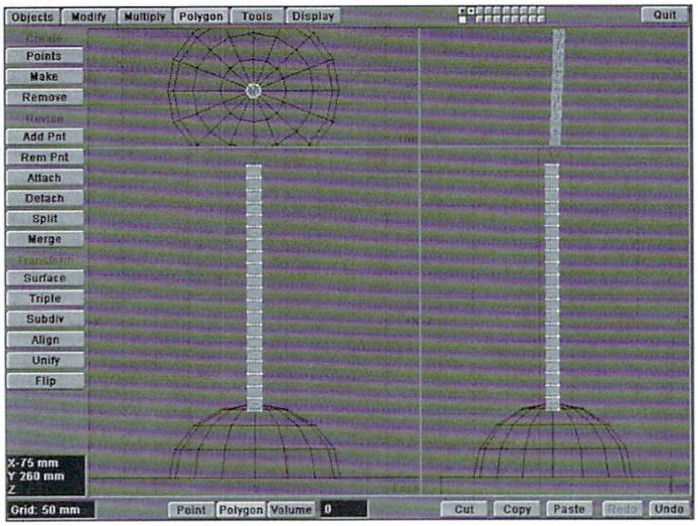


Figure 11: Tripling polygons in preparation for use as a morph target.

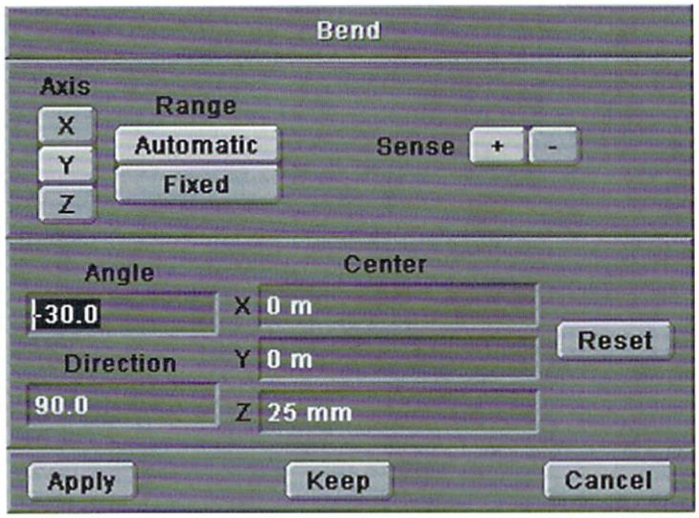


Figure 13: The Numeric requester for the Bend tool.

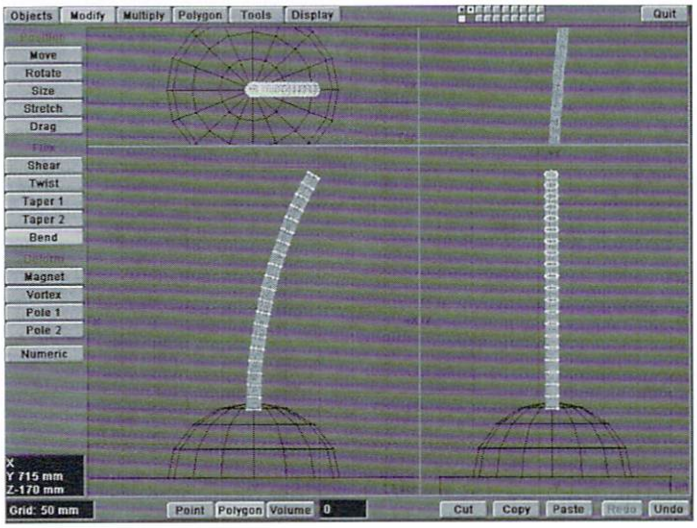


Figure 12: Bending the antenna object.

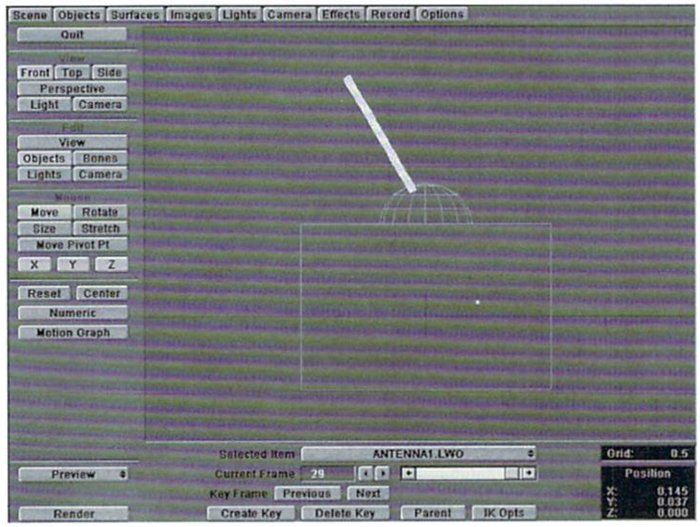


Figure 14: Positioning the antenna on the TV set.

ACT II

- Returning to Layout, Load all the objects we just created—the TV object and the three antennas. From the Objects panel, make both Antenna2.lwo and Antenna3.lwo 100% dissolved.
- With Antenna2.lwo as the current object, specify Antenna3.lwo as the Metamorph Target. Click the Envelope button (E) next to the Metamorph Amount requester to bring up the envelope graph. Set two more keyframes using the Create Key button (enter), at frame 4 and frame 8 (there will already be a keyframe at frame 0). For keyframe 4, enter a value (in the requester at the bottom of the envelope chart) of 100%. Open the Spline Controls panel (s) for all three frames and click on Linear for each of them. Finally, make sure that the envelope End Behavior is set to Repeat (default is Stop). You should have an envelope like the one in Figure 3. Click Use Envelope to keep these settings.

Our two antennas are now oscillating between each other. Of course, we can't see anything yet because we've made both of them 100% dissolved.

So let's continue:

•Select Antenna1.lwo as the current object and parent it to the TV set object. Once that's done, use the Front and Side views to move the antenna to its place on the antenna stand (Figure 14). Keyframe this, making sure, of course, that your current frame is 0.

- In the Objects panel, select Antenna1.lwo as the current object and Antenna2.lwo as its Metamorph Target, and then click on the Envelope button (E) by the Metamorph Amount requester.
- In the Envelope graph, create the following keyframes:

  Frame 30, Value 0%, Spline Tension 1.0
  Frame 32, Value 100%, Spline Tension 1.0
  Frame 100, Value 0%, Spline Tension 1.0

Your envelope should look like the one in Figure 4. Click Use Envelope to accept these settings.

- You've just completed one antenna on the television set. To duplicate this process for the other antenna, simply clone Antenna1.lwo from the Objects panel, rotate its heading 180 degrees and move it into position on the other side of the antenna base. Don't forget to keyframe it (Figure 15). Note that this cloned object also has Antenna2.lwo as its Metamorph Target. This works just fine, and there is no need to clutter up your scene file with a different copy of a metamorph target object.
- From the Scene panel, adjust the Last Frame number of your scene to 110 or so. Keyframe a good viewing angle for your camera, save your scene file and make a wireframe preview.

Notice how the antennas start vibrating at around frame 30, looking like they've been hit, and then slowly stop vibrating? All you need to do is keyframe

see Antennas, page 16

Makin' Tracks

Creating Working Tank Treads

by Joe Angell

For one of my personal projects, I needed a tracked anti-aircraft gun to menacingly roll over some terrain. The problem I had was figuring out how to get the tank's tracks to roll at all. After tinkering around for a few hours, I finally came up with a convincing way to make turning, rolling, terrain-hugging tank tracks.

Start Treading

The first step is to make the tracks themselves. I'm going to go ahead and assume you've already built the rest of the tank. *[Editor's note: Check out tomahawk.welch.hu.edu, an FTP Internet site containing various objects.—JG]* By that I mean you have the tank body that the guide wheels attach to and the guide wheels the tracks wrap around. You should also have a scene file set up with the guide wheels parented to the tank's hull and configured appropriately.

- To figure out how long to make the tracks, you'll have to Measure (Options panel) in Modeler around the wheels that the track will wrap around. Set up your tank in Modeler the way it would be in Layout, with the guide wheels in place on the body. Then measure around the wheels. I took four measurements around my wheels (top, bottom and sides) and added them up to get 7.1 meters. This measurement will be the length of the track.
- Select Box (Object panel) and hit (n) to bring up the Numeric options. Set the number of Segments on the Z-axis to 100. The Z-axis size should be set to the length of the track, which you got in the last step (in my case, 7.1 meters). The width should be slightly more than the width of your wheels, and the thickness just a centimeter or two.
- You may want to add some extra details to the tracks. I did a low Bevel (b) (Multiply panel) off of every other polygon from the Top view to get a tread effect. I also did another, lesser bevel on the bottom of the tracks for yet more detail.
- Save this object as "Track-Start". Move the track the length of two segments (polygons) on the +Z-axis. Save this again as "Track-Morph". An example of the slightly offset track is shown in Figure 1. What you've just made is a flat tank track and a copy of the track moved a slight bit on the +Z-axis. This will be used to move the track forward with a morph.

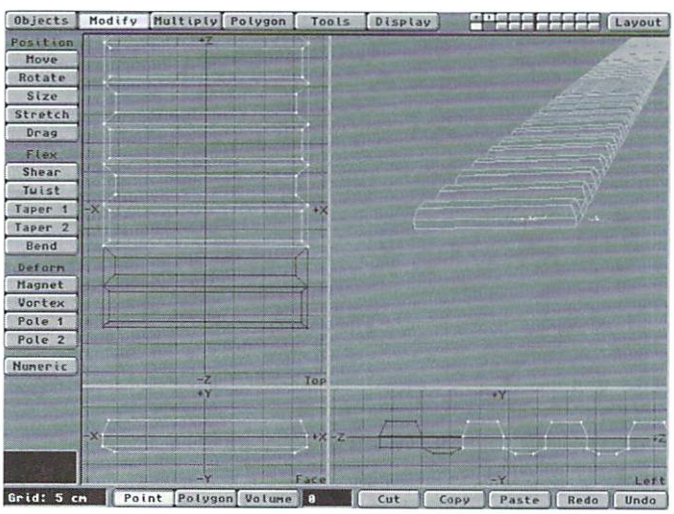


Figure 1: Track-Start (in Background Layer) and Track-Morph (in Foreground Layer) as an example of shifting the track two segments.

Moving By

- Go into Layout and load up a scene that has the tank and its wheels set up. Load both the Track-Start and Track-Morph objects. Parent the Track-Start object to your tank's hull, not the wheels, since you'll want them to spin separately from the track. Move Track-Start so it is resting over the wheels, with the "back" end about midway between the front and rear tank wheels.
- Go into the Objects panel and set the Object Dissolve for Track-Morph to 100%. Set the Track-Start's Metamorph Target to Track-Morph. Track-Morph is there only as a template to morph into, so it will never be seen. Now the whole morph is set up except for the envelope.
- By editing the Envelope (E) of the Metamorph Amount, the track will move forward or backward. It will only move two segments of track, but once we're finished, it will be a seamless transition. To make the track go forward, have the morph percentage start with 0% and go up to 100% over the number of frames you want. To go backward, start from 100% and end at 0%. You will want to set the envelope's End Behavior to Repeat. (This will all make more sense later on.)

A track that moves forward and jumps back to the beginning over and over again is useless unless it wraps around the wheels of the tank. That's where Bones come in.

Dem Bones

Bones are fairly painless in this case. All you need are a bunch of them. You don't even have to mess with any of the settings in the Bones panel. We're going to use bones to bend the track around the wheels.

The way this works is a bit strange. You will want to arrange your bones along the track in their rest position at Frame 0, and move your bones at frame 1. This way it'll be easier to add more bones for greater control later.

- Go into the Objects panel and select the Track-Start object, the object you'll be adding all the bones to. Select Object Skeleton and click Add Bone. Hit (p) to close the Bones panel and go back to the Layout screen. This first bone is going to be used to "hold down" the rear end of the track, so you'll want to position it there. Place the bone so the pivot is in the track and the other end is pointing upward. If you think of the bones as the control rods of a puppet, this setup will seem a little clearer. Those of you who are more technically minded can think of these as the control handles of a B-spline.
- Add two more bones by hitting the (+) key twice. Do not use Add Child Bone, as this will only complicate things. Position Bone 2 so it is over the

frontmost wheel. Bone 3 should go further beyond Bone 2 by an amount equal to a diameter and a half of the front wheel. Don't forget to hit (r) to set the Rest Position and activate each bone after setting its position and creating a key at frame 0.

- Now go to frame 1. Take Bone 3 and rotate it 180 degrees so it is pointing downward. Drag the bone down until it is under the front wheel. The track should now be more or less bent around the front wheel. Notice how this doesn't quite work. The track is a little bit off the front of the front wheel. Now you know why we are doing all of our setup on frame 0 and our final positioning on frame 1.
- Go back to frame 0 and hit (+) to add another bone. Position Bone 4 so it is between Bones 2 and 3. Create a key for it and hit (r) to set the Rest Position. Go to frame 1 and you'll notice that the track is warped severely. Rotate Bone 4 approximately 90 degrees and place it slightly ahead of the rear wheel. This extra bone is used to keep the track against the front wheel.

Enough With the Bones

We've covered the basics of setting up the tank track. To get it all working requires a fair number of bones—I used 18 for mine. For an idea of what the bones look like after they are all oriented correctly, Figure 2 shows frame 0 and Figure 3 shows frame 1. Pay attention to the front of the tank (on the right of the screen) to see how those first three bones are set up.

- If you need to adjust a bone (and you will), you'll want to move it at frame 0 and set the Rest Position again. If you move the bone at frame 1, the track warps as the bone stretches it. If you first change the positioning on frame 0 and reset the Rest Position, you can safely move it the same distance on frame 1 without any nasty warping effects.
- There are two other important bone placements. The bone at the rear end of the track on frame 0 should be moved on frame 1 so it exactly matches the first bone's position on frame 1. That way the entire track smoothly wraps around the wheels.

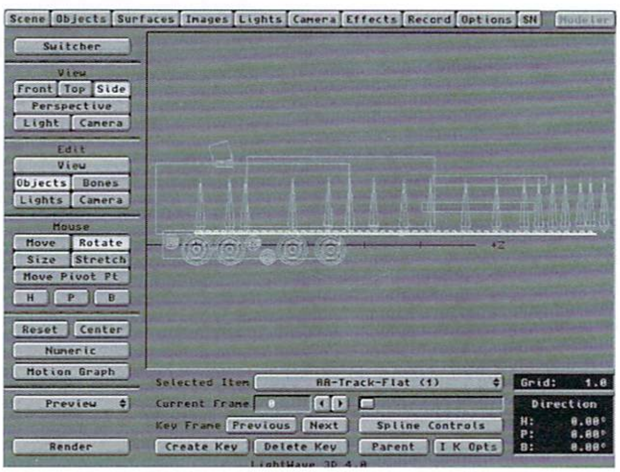


Figure 2: Frame 0 of the treads with the bones positioned.

- The final trick is to put a last bone about two track segments (yes, we're back to that again) ahead of the frontmost bone, so it is actually off the end of the track. This should also be moved forward two segments past the last bone on frame 1. The reason for this bone will make sense in the next step, when we get this thing moving.

Did I say bones were painless? I lied.

Let's Make Tracks

Remember that morphing setup we did way back when? Now we're going to use it.

- Go to the Metamorph Amount envelope for Track-Start.
- For a test, set frame 0 to 0% morphed and frame 10 to 100% morphed. Make sure to set the End Behavior to Repeat.
- Now go ahead and make a wireframe preview.

When you play this back, the tank track should be smoothly turning around the tank's wheels. The reason for putting that last bone just forward of the front end of the track is to keep the track from warping as it morphs. Without that extra bone, the track starts bending upward, which ruins the illusion.

By now you've probably noticed a major limitation with this method: texture mapping. Since the track only moves two segment's worth, the texture map has to be two segments long and must be seamless, or else there will be a noticeable jump in the texture as the morph repeats. With a little work and some clever mapping, you can get around with a little work and a program like DeluxePaint.

I Failed Math

To control the speed of the tracks, you simply have to adjust the number of frames the morph occurs over. Add more frames and the tracks move slower. About five minutes of tinkering is all you need to synchronize the turning of the tracks to the turning of the tank's wheels. If you want to be scientific about it, try the following formula:

r = radius of guide wheel
l = length of track
d = distance to move track per full guide wheel rotation

$$\frac{\pi r^2}{l} = d$$

In other words, if you divide the circumference of the wheel by the length of the track, you get the distance the track should move for

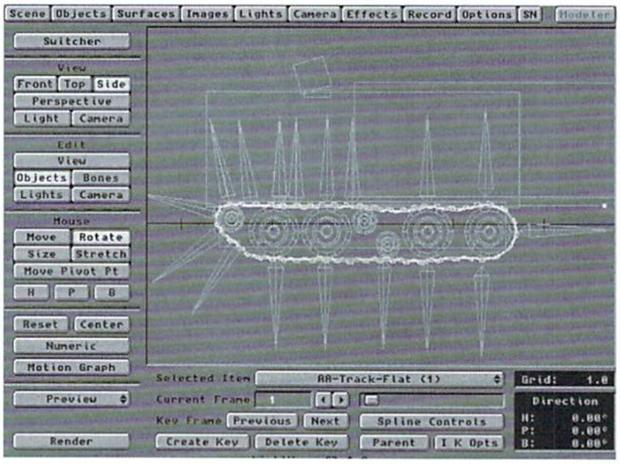


Figure 3: Frame 1 of the treads with the bones positioned.

each full (360-degree) rotation. You will have to measure the distance of two segments worth of track in Modeler, and divide that by the distance to move the track. Then you will have the number of frames to morph the object over. As for an equation:

d = Distance to move Track per full Guide Wheel rotation

s = Distance of two segments of track

$$\frac{d}{s} = \text{Number of frames to morph over}$$

Remember to use meters for all of these distances! You can also use Joe Dox's acceleration and deceleration macros (from *LIGHTWAVEPRO* disk 10) to control the rotation of the wheels and to figure out the amount to morph the tank tread. If I ever learn ARexx, I'll have to write a script for this.

Finishing Touches

One of the great things about using bones for this project is that you can have the tracks bend as the tank passes over different terrain. As it bumps over a rock, the track can compress against the wheel, and as the hull hangs in the air, the track can become more slack.

Making the track for the other side of the tank is easy. Just use Clone Object on the first track. The morphing information and bones are all kept during the clone, so all you need to do is move the track over to the other side of the tank.

This method of generating tank tracks gives you an incredible amount of control over making the track move and react to the terrain. Once you get everything set up, it's simple to control. Now all you need to do is build the terrain!

LWP

Joe Angell is a film/animation/video student at the Rhode Island School of Design. When he's not in class, he tries to actually get some CGI done. E-mail him at jangell@RISD.edu.

Antennas

continued from page 13

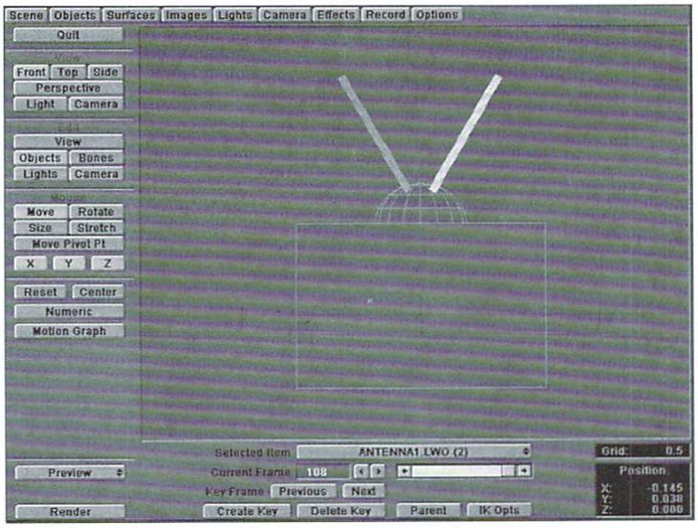


Figure 15: Cloning a second antenna object.

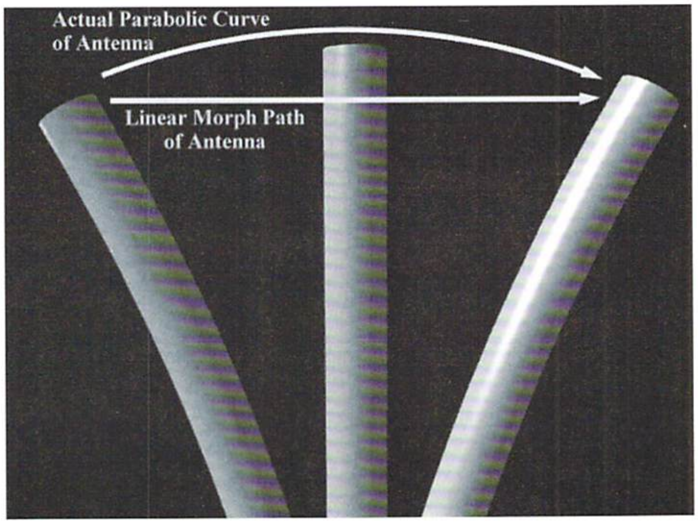


Figure 16: The problem with linear morphs.

the TV set so it falls and hits the ground (the motivation for the vibrating antennas) at frame 30.

ACT III

- Select your television object as the current object. Enter the following keyframes and values for it:

Keyframe 0:
X, Y, Z = 0, 2.687, 0
HPB = 0, 0, 9.7

Keyframe 30:
X, Y, Z = 0, 0, 0
HPB = 0, 0, 0
Spline Continuity = -1

Keyframe 35:
X, Y, Z = 0, 0.057, 0
HPB = -0.7, 0, -2.8

Keyframe 38:
X, Y, Z = 0, 0, 0
HPB = -1.1, 0, 0
Spline Continuity = -1

Keyframe 41:
X, Y, Z = 0, 0.018, 0
HPB = -1.6, 0, 0

Keyframe 45:
X, Y, Z = 0, 0, 0
HPB = 0, 0, 0

(Values may vary depending on the size of your TV object.)

- Make another wireframe preview. You will now see the television set fall and hit the (nonexistent) ground, which causes the antennas to vibrate.

EPILOGUE

You may have already noticed from doing this tutorial that if you bend the antennas too much while modeling them, your morph may not look too good. This is due to the fact that our morph is a linear morph. If you track a point on the top of an antenna through the morph, you'll see that it moves in a straight line toward its target. In reality, the same point on a real antenna would follow a parabolic path to the other side (Figure 16). In our example this error is small enough that it isn't noticeable, and the fact that it vibrates at a fairly high speed also helps to cover up the error.

If we bend it too far, however, the linear morph is pulled too far from the parabolic path and the antenna structure will begin to break down. In instances like these, you may place additional morph targets. Be aware, however, that each additional morph target means having to coordinate between morph envelopes, which can easily get out of hand as the number of morph targets grows.

This issue also brings up timing. In the last part of the tutorial, you blindly entered the keyframe values for the television. Where did these numbers come from? How were they derived? These, of course, are questions for another article.

In the meantime, what will happen to CGIMan? Tune in next time for the shocking conclusion.

LWP

Alan Chan is now a professional Ridge Racer and air combat veteran in between Boolean operations. His pit stop area is achan@ix.netcom.com.

Modeler Mysteries

continued from page 5

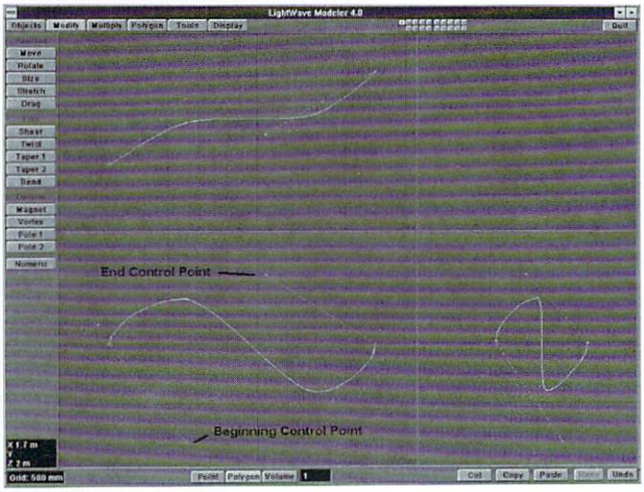


Figure 4: This curve has both a starting control point and an ending control point activated. Notice that the actual curve starts in from the control points.

and the start point hops from one end to the other. The start point will be important when you are patching.

You can move the start- and endpoints in a curve "down the line" (in the case of a startpoint) or "up the line" (in the case of an endpoint) by selecting Start CP (Ctrl–b) or End CP (Ctrl–n). This is useful when you need the startpoints or endpoints of a curve to maintain a smooth curve, since the first and last points of a curve always "terminate" the curve. When you create start or end control points, you will see dotted lines connecting the new startpoint or endpoint of a curve to the "control" points (see Figure 4). The control points aren't actually parts of the curve but rather allow you to modify or control the end of your curve. Drag the control points around and watch the results.

You can also create a curve by selecting a series of points and clicking Make (Return) under the Tools menu or pressing (Ctrl–p). For a closed curve, one with connected ends, hit Make Cl (Ctrl–o). Using Make Cl automatically produces a smooth closed curve by using points in the curve as the start and end control points.

LWP

Tom Williamson is the effects supervisor for Santa Maria-based Computer Café. He's about 5'8", drinks a lot of Coke and is gonna be a dad in February 1996.

COMING IN Future Issues of LIGHTWAVEPRO

- **Everything you always wanted to know about ScreamerNET**
- **Performance-improving tips for Windows NT and LightWave**

What's on the Disk:

This month's *LIGHTWAVEPRO* disk includes object, image and scene files from "lwpro@internet.online," "Mighty Morphin' Television Antennas" and "Makin' Tracks."

Tip of the Month:

The final shipping version of LightWave contains three hidden features which can be accessed by some keyboard combinations. All are activated by first pressing Ctrl+Shift+F1 in Layout (no panels open). You will get a message telling you that experimental features are activated. After pressing this combination, you can do any (or all) of the following: (1)MTSE morphing (Multiple Target, Single Envelope—Enter the Objects panel. Press Ctrl+Shift+F2 when you have the main source object selected. Now the Morph envelope for the source object can be used for *all* targets. The envelope from 0%–100% is for the first target, 100%–200% is for the second, etc. Make sure that your subsequent targets all have their respective Metamorph Targets set up correctly. If you want A to morph to B and B to C, you would set up your targets as usual, but only use an envelope, for the first source/target combination.(2) Cell Look—Enter the Objects panel. Press Ctrl+Shift+F3 for the currently selected object. Now, the object will be rendered with an outline. The outline color is determined by the Polygon Edge color; the thickness is determined by the Particle/Line Size value. (3)Field Mapping—Enter the Images panel. Select Ctrl+Shift+F2 for the currently selected image. LightWave will now use "field mapping" for this image. Like Field Rendering, you will be basically be mapping 60 images per second rather than 30 (at NTSC rates).

If your images or sequences are not field rendered or captured video, you won't see any advantage. Disable these features by pressing the same key combination for each option or pressing Ctrl+Shift+F1 in Layout again. You will always get messages telling you if a feature has been enabled or disabled.

BUSINESS REPLY MAIL

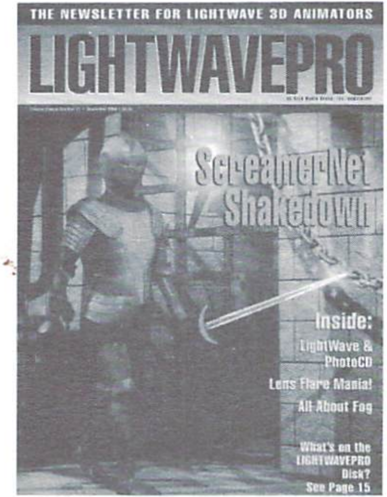
FIRST-CLASS MAIL PERMIT NO. 2263 SUNNYVALE,CA

POSTAGE WILL BE PAID BY THE ADDRESSEE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

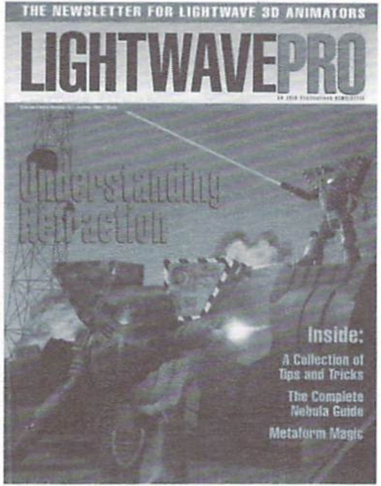
LIGHTWAVEPRO

273 North Mathilda Avenue
Sunnyvale, CA 94086-9313



TO ORDER WITH YOUR VISA
OR MASTERCARD
CALL TOLL FREE!

1(800) 322-2843



TO ORDER WITH YOUR VISA OR MASTERCARD CALL TOLL FREE!

1(800) 322-2843

LIGHTWAVEPRO

THE NEWSLETTER FOR LIGHTWAVE 3D® ANIMATORS

Make a Smart Investment

Subscribe Today!

Canadian/Mexico: Add $US12

Overseas: Add $US36

Allow 6-8 weeks for delivery.

Make checks payable to LIGHTWAVEPRO.

Prepayment required on all overseas orders.

YES! Enter my one-year (12 issues) subscription to LIGHTWAVEPRO at the Special Introductory Rate of only $48—that's 50% off the cover price!

Name ______________________________

Address ______________________________ Apt. # ______

City ____________________ State ________ Zip __________

☐ Payment Enclosed ☐ Bill Me

BFXLW

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

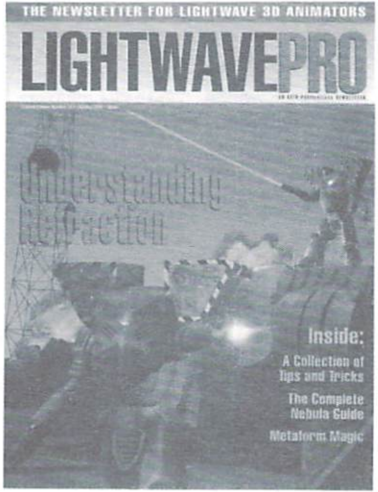
BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 2263 SUNNYVALE,CA

POSTAGE WILL BE PAID BY THE ADDRESSEE

LIGHTWAVEPRO

273 North Mathilda Avenue
Sunnyvale, CA 94086-9313



TO ORDER WITH YOUR VISA
OR MASTERCARD
CALL TOLL FREE!
1(800) 322-2843


So Fast
It's Scary...
It had to happen. The fastest Windows NT workstation was starving for the killer application. LightWave 3D had a need for speed. Animators were looking for the complete solution. NewTek Systems Group brings them all together. The NewTek Raptor provides five configurations with ultimate upgradability. Uniflex architecture allows new processors to be added to the system without sacrificing your investment. LightWave 3D is the most powerful 3D software available for any platform, especially with the brute force of an Alpha processor.
Just ask Hollywood. Amblin Imaging, Area 51, Foundation Imaging, and many other studios use LightWave on Alpha workstations to produce thousands of frames every week. So when you're ready to get serious about animation, get LightWave, get a Raptor, and get to work.
MULTIMEDIA WORLD ALL-STAR
NEWTEK.
Raptor™
BEST OF BYTE COMDEX Spring '95
For information call NewTek, 1-800-847-6111, Outside the U.S.A. call 1-913-228-8000, FAX 1-913-228-8099
Consult your dealer to configure a system to meet your needs. Specifications subject to change without notice. Newtek Raptor is a trademark of the Newtek Systems Group. Raptor is a trademark of DeskStation Technology. LightWave 3D is a trademark of NewTek, Inc. Windows NT is a trademark of Microsoft, Inc. Alpha is a trademark of Digital Equipment Corp.