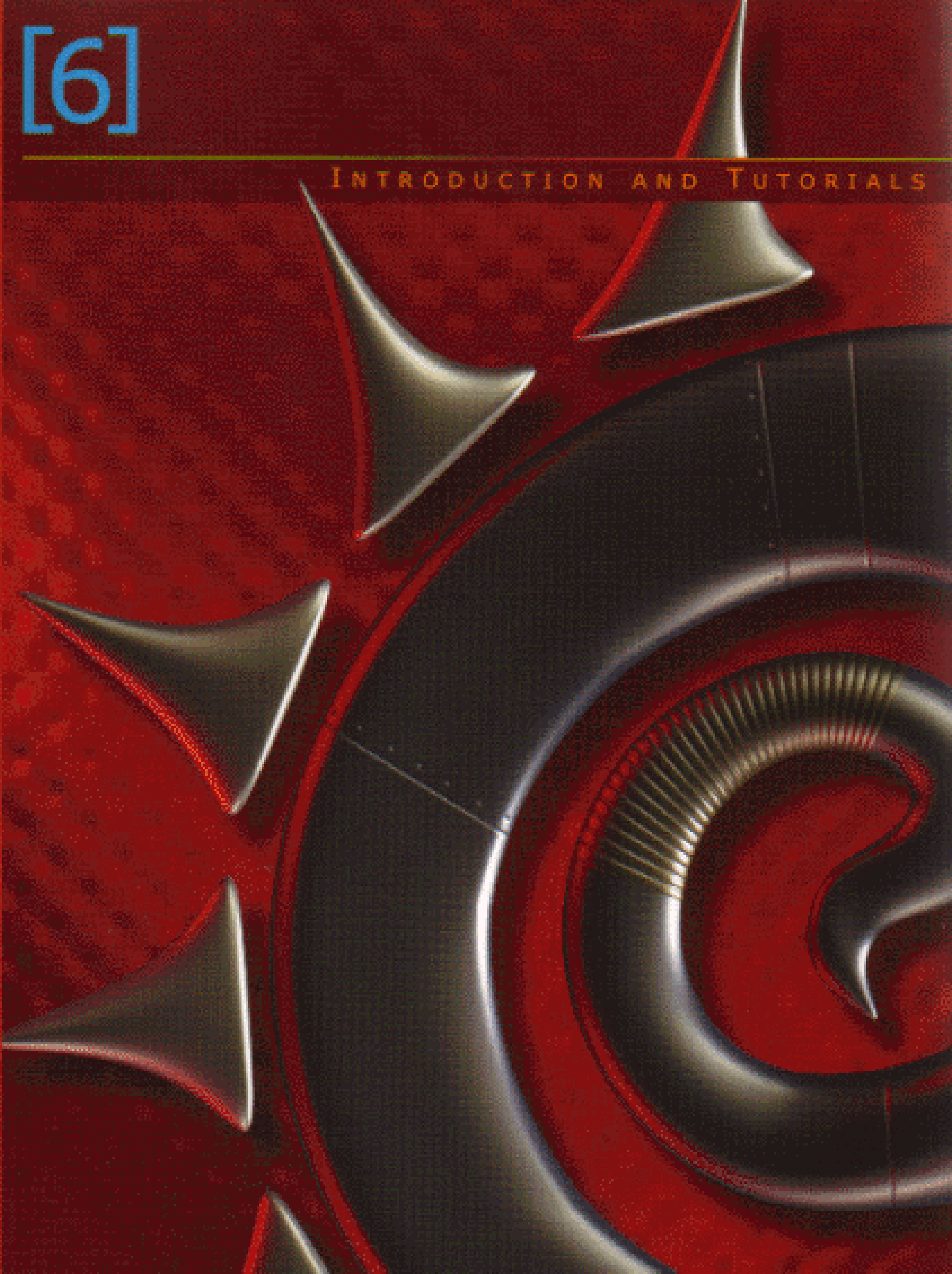
LightWave [6]

Introduction and Tutorials



chapter 1
Introduction

CHAPTER 1: INTRODUCTION

Congratulations on your acquisition of LightWave [6]. We believe that you will find using this software as rewarding as we found developing it. Many of you are long-time LightWave users and many of you are new entrants into the LightWave community. Something that you will find immediately obvious is that there is something different about LightWave!

WELCOME ABOARD

For you long-time users you will notice that while the system looks familiar things are mighty new. The interface is streamlined, the architecture is overhauled and the program just generally feels beefier. To minimize your transition period we have included a section specifically for LightWave 5.5/5.6 users. We recommend blasting through that section to get acclimated before you go full on into the application. Of course we realize you have already been playing with it before you even opened this book!

The new users will notice many things that are unique about LightWave. First of all the interface may seem unusual to you. We have spent the last 11 years developing a human interface for 3D graphics and animation work and have found there are many unique problems to solve when presenting the level of technological power contained in this application to users who would rather focus on art than technology. Don't feel bad if things frighten you a tad. You will soon find the interface vanishes from before you and leaves you with an almost direct conduit to your creative vision. This interface is designed for productivity and while it looks a little different at first you will find that it is a streamlined environment that proves to be a joy to work in. Not only do our programmers and development managers create 3D tools, we use them. All of us on the development team are also animators. So we know what it is like to sit in front of our program for hours on end tweaking out a shot to get it just right. In fact most of got into this by making animations first, getting fed up with the quality or expense of other software and just deciding to do something about it!

COMMUNITY

Another unique attribute about LightWave is the LightWave community. It is rare to find such a fun loving bunch of hooligans that are so talented and yet so willing to share their secrets, ideas and creations. Make certain you take a look into the available resources such as the LightWave Mail list, Newsgroup, local users groups, training videos (www.DesktopImages.com), and NewTek specific magazines (http://www.keyframemag.com/) (www.NewTekniques.com). For up-to-date information, visit www.NewTek.com and look in the tech support section.

ARCHITECTURE

LightWave [6] is truly a powerful piece of software. This program represents over 10 years of research and development from a team of insanely dedicated programmers, artists and visionaries. Perhaps the most exciting aspect of this new software is the architecture upon which it was developed.

There are many aspects to this new architecture that allow for an incredible amount of power, flexibility and expandability of this software. Some of the components that make up the new architecture are: cross-platform tools, expansion hooks, data formats, modeling engines, texturing tools, animation engines and render pipeline. Each one of these subsystems adds a tremendously valuable piece to the application.

CROSS-PLATFORM SUPPORT

Lightwave 3D has always been the most cross-platform friendly, high-end 3D animation program available on the market. At NewTek, we feel it is important to let the artist have their choice of platform. We also understand that many users will work with one operating system at work and a completely different OS at home.

We realize that animators spend a lot of time in our software. It becomes almost a second home for some! Therefore, it is important that users of LightWave can feel comfortable inside LightWave regardless of what operating system they are using. With LightWave [6], we have gone to great lengths to ensure that the user experience is the same no matter what platform you choose.

If you use NT at work and a Mac at home, you should have no difficulties switching back and forth. We have ensured that data is compatible across systems and that the user interface is the same no matter what system you are running. We have also isolated platform specific code so that the majority of LightWave on any platform comes from the same code base.

It also guarantees that each platform has a native compiled application running rather than a simple OS emulation or quick port. We even have platform teams that can optimize our code for different platform specific technologies as they arise, such as MMX, SIMD or Altivec technologies.

INFORMATION FOR MACINTOSH USERS

A RMB action (i.e., drag or click) is accomplished by holding the COMMAND (APPLE) key while pressing the mouse button. Also, to change the amount of memory allocated to LightWave 3D, select the icon and choose **File > Get Info**.

EXPANSION

As the field of 3D animation has matured, we have had to increase our development pace to keep up with user needs and to stay ahead of competitive forces. As such, we realized that a critical aspect of this release of expandability.

In the design of this program, we took into account the need for us to expand our software rapidly in the future, as well as allowing maximum expandability for third parties and end users directly. LightWave [6] allows expansion of the system through a number of mechanisms.

At the application level, NewTek has created a data-sharing repository called the *HUB*. The protocol for attaching to the HUB is public domain. This will allow other applications to share data with LightWave in a live fashion.

At the next level, there is an expansive SDK (software developers kit) with an advanced API (application programming interface) that allows third parties to create C and C++ code that can be added into the system as modules. This SDK/API combo has been updated from LightWave 5.6 to address many needs and desires of third-party developers. With this powerful new system, many exciting features (even not so exciting features!) can be added to the system very rapidly.

Finally, there is an embedded scripting language in LightWave [6]. Building the entire application *on top* of this command sequence system allows a vast amount of scripting control. By leveraging this new command sequence engine and the power of LightWave's current scripting language, LScript, end users have a simple to use, but extremely powerful mechanism for adding features through the use of macros and scripts. This will make it possible to automate redundant tasks or create your own high-end features.

DATA FORMATS

Some of the most exciting aspects of LightWave [6] come from the new data formats. LightWave [6] has a completely new scene file definition (LWS3), a powerful new object format (Intelligentities) and new image file formats (flexible precision and high dynamic range images). These new formats allow many things.

The scene format now provides a mechanism for animation to be based on independent channels. It also allows for floating-point entry of all variables in the scene for IEEE precision as well as high dynamic range values for lights and other elements in a scene.

While developing LightWave [6], we knew that one of the most important areas for improvement in the field of animation is character work. It is incredibly difficult to bring a digital character to life and perhaps the most overlooked aspect of this task is setting up a character. Once a character is set up properly, animating the character becomes an artist's endeavor, rather than a technician's struggle.

With that in mind, we added many new data components to our object format to allow us to simplify the character animation process. Some of the simple additions include *vertex maps* for assigning point clusters for bones or selections sets based on polygon or point groups.

The same vertex-map technology can be used to effect the various modeling tools by acting as a modifier to the tools strength. In addition, vertex-map can be used to change the contouring of our *SubPatches* for adding creases or tight edges to otherwise organic objects. These vertex-map can also be used by third parties for assigning characteristics to geometry for dynamics engines, particle systems, or for direct integration to a real-time gaming engine.

Another unique form of vertex-map is an *EndoMorph*. An EndoMorph is a method for assigning alternate locations to points in the model (i.e. morph targets). By defining an EndoMorph in the object, users can essentially *train* the model for different poses that can be animated or blended with other EndoMorphs in a completely non-linear fashion. This is a tremendous tool for facial animation and lip sync.

One major benefit to this EndoMorph system is the ability to make changes to the point/polygon count of your model. Most morphing systems require that the base (or anchor) model have the exact same number of points and polygons as the targets. This means that once you have started building targets you can no longer add or subtract geometry from the base without making the same changes to all the targets you have created. This can be a daunting proposition.

With EndoMorphs you can make changes to the base geometry and these changes will be propagated through the endomorph maps. EndoMorphs are a fantastic way to create subtle and intricate morphing animation and blending without the hassles associated with traditional morphing systems.

With MultiMeshes LightWave [6] can store object data in unlimited layers, similar to image layers in paint applications like Aura Paint or Adobe Photoshop. These object layers can then be separated for further editing, or animated as individual elements. Each of these multi-mesh layers can have their own set of EndoMorphs and other vertex-map data.

The new object format is also very expandable for third-party developers and because it is an openly documented format, anyone can use it. Most 3D applications keep their data formats proprietary and limit developers' exposure to the data set. NewTek believes this sort of information should be widely available so that applications can share data and third parties can extend and improve the application more effectively. Robert Landsdale of Okino Graphics, makers of PolyTrans, had this to say about the new format:

> *Since 1995, I have been working to create a perfect Lightwave import and exporter for our PolyTrans Data Translation software and our NuGraf Rendering System. With its good Lightwave support, PolyTrans is a popular utility to move data to/from Lightwave, such as for game development. To date this has resulted in well coded and maintained I/O modules which handle most of the important aspects of the .lwo and .lws files, including all mesh data, materials, hierarchy, lights, cameras and animation.*
>
> *However, I have long lobbied for UV texture coordinate support in the .lwo format, the ability to save out more than 65,536 polygons/vertices to a single .lwo file and the ability to animate individual channels (for example, just the x,y, or z translation channels) instead of all the channels at the same key (a limitation of the old .lws file format).*
>
> *I am very happy to see all these changes made in the new .lwo and .lws file formats. Newtek also took the right approach of extending the old .lwo/.lws formats instead of completely replacing them (my 5 years of .lwo/.lws converter development has not gone to waste), or worse yet, making them undocumented and proprietary formats as other 3D vendors have chosen to do lately.*
>
> *Keeping the .lwo/.lws formats open and documented will ensure the continued popularity of Lightwave to game developers who often need to gain access to the raw Lightwave file data.*
>
> *–Robert Landsdale*

MODELING TOOLS

LightWave 3D has always been well known for the power of its modeling tool set. As a native polygonal editor it has one of the most robust arsenals for manipulating polygons en masse or individually. LightWave's modeling tool set even allows users to edit a mesh at the vertex level with numeric precision. At the same time, users can edit meshes visually with interactive tools that give instant feedback.

While these polygonal editing tools are extremely powerful, it is important to have a more organic solution for creating smooth flowing shapes. LightWave charted the course for the 3D graphics industry by providing users with a comprehensive set of tools for working with Subdivision Patches in 1995. This technology was only recently adopted by most of the industry in more recent years and is now the de facto standard for creating organic models.

This paradigm provides an organic solution to modeling with the ease of use of polygonal editing. LightWave [6] also has support for real-time metaballs modeling, as well as traditional spline patching tools. Of course, all of modeler's tools have been optimized to be more interactive and provide instant feedback for the user.

TEXTURING TOOLS

The texturing tools in LightWave [6] are quite impressive. There are many new texture attributes such as translucency, glossiness mapping and refraction mapping. LightWave [6] also has three texture types that can be applied and layered to any of the surface attributes. These texture types are Image Map (6 types), Procedural textures (16 types), and Gradients with various input parameters.

These texture types can be infinitely layered with seven different blending modes and variable opacity. When layering textures, the order of textures can be changed in a simple drag and drop interface and layers can be copied and pasted from one surface to another with the just a couple clicks of the mouse.

To assist in the management of surface heavy scenes, there are also a number of filters that can be applied to limit the number of surfaces appearing in the list. It is also now possible to add text comments to a surface, so that when working in a group environment, it is possible to keep track of the status or changes needed for a particular surface or texture.

Another tool for assisting in the management of surface editing is the ability to use the Versatile Interactive Preview Render window (VIPER), discussed later, to simply click on a rendered image to pick the associated surface. This allows users to look at the rendered scene, click on the image and quickly edit the surface. Using VIPER, the changes are made to a sample sphere or cube, but most changes can also be seen immediately on the final rendered image.

Once a surface is completed it can be checked into the Preset Shelf for later retrieval. With the Preset Shelf, surfaces can be easily retrieved with by double clicking the appropriate rendered sample image. Work sessions can be created to help manage complex lists or groups of surfaces and the very same presets shelf can be used in HyperVoxels, volumetric lights and even by plug-ins that support the shelf API. This is a fantastic method for visually organizing and tracking your content.

ANIMATION ENGINES

LightWave [6] has many new tools for animation, but the most important are the new engines underneath it all. Those include a soft-body dynamic engine, real-time 2D/3D inverse kinematics engine, new real-time particle engine, EndoMorphs and PAVLOV (discussed later). Combined with all the new animation features leveraging these technologies, LightWave [6] is a formidable animation powerhouse.

SOFT-BODY DYNAMICS

With the inclusion of Motion Designer for LightWave [6], users have access to an extremely sophisticated dynamics simulator that can apply forces such as collision, wind and gravity onto soft body surfaces such as clothes, skin, hair and even liquids.

INVERSE KINEMATICS

All good animation systems are based on their IK system. LightWave [6] has one of the most sophisticated, fastest, most accurate and flexible IK systems available on the market. Unlike many systems, this engine can be used to solve completely three-dimensional IK chains with multiple goals per chain.

The same IK engine can be restricted to calculate only in a 2D fashion (often more appropriate for character joints). This new system also allows the user to keyframe certain axes while using IK to solve the others. This level of flexibility is essential for the rising demands in the animation process. With this system in place, there is no limit to the types of IK scenarios that can be solved. At the same time, this new IK system is much faster and more accurate than the renowned Lightwave 5.6 IK engine.

ENDOMORPHS

To assist in the creation of facial animation and lip sync LightWave [6] has added EndoMorphs (see also the previous discussion). Once these EndoMorphs are added to an object they can be animated through the EndoMorph Mixer panel or by modifying channels in the Graph editor.

The EndoMorph Mixer is a floating palette of sliders, one slider for each independent EndoMorph. Users can quickly animate these values by advancing to a frame and adjusting the EndoMorph sliders. This provides a simple and fast method for creating lip sync and other multi-layered morph animations. For more precise control of the EndoMorphs users can activate the Graph Editor and directly manipulate the channel for each EndoMorph. These curves can be modified independently or in groups. This also provides a mechanism for attaching channel modifiers to EndoMorphs for functionality such as adding noise to a channel or simply locking an EndoMorph to some other value in the Scene.

PAVLOV

Parameterized Animatable Values Linking Objects and Variables (PAVLOV) is a central repository for all animatable values in a scene. With LightWave [6], nearly all values can be changed over time with an envelope. All of these envelopes are stored as common channel data inside the PAVLOV engine. By storing all channels in a common form they can be linked or referenced to each other. Envelopes from different elements can also be edited simultaneously and even have a texture applied to them.

PAVLOV also provides for an expressions engine so that these values can be changed with mathematical functions. Common uses for expressions are to link the position of one item to always be centered between two others, as is the case with hips between two feet. Although expressions require a bit of up-front legwork, they can be a tremendous time saver over the course of an animation.

Another important aspect of PAVLOV is the Gradient function. Gradients allow many different surface attribute values to be linked to other parameters (parameterized) of the scene. Some common examples are changing the color based on the bump height of an object or varying the transparency based on the incidence angle to a camera or light.

With PAVLOV, many aspects of an animation can be automated. For instance, one could establish a relationship between light intensity and cone angle such that opening a light's barn doors would automatically increase the brightness. Even more complex scenarios can be devised such as linking the rotation of a wheel to the position of the car. It is even possible to create a PAVLOV link such that rotating a single bone would trigger an EndoMorph to flex a muscle which would in turn increase the bump intensity of a wrinkle map around the elbow joint. There is no end to the fun you can have with PAVLOV. Taking a little time up front to create these relationships will pay off in the end. But isn't that always the case?

THE GEOMETRY PIPELINE

In order to more fully utilize this application it is important to look past the buttons and understand how the application functions internally. With any software program, an in-depth understanding of its mechanisms will allow the user to more completely push the software. There are many areas of this application that need not be explained for casual use, but once understood will unleash a new level of user satisfaction. These areas include the geometry engine, PAVLOV, the animation pipeline, and the rendering pipeline. By taking the time to understand these systems, you will be able to *bend the rules* and take LightWave [6] places even we didn't intend! Such is the domain of the power user.

There are some simple things to understand about the geometry engine. *GCore* (Geometry Core) is the *engine* that sits underneath all animation and modeling tools. This is the portion of the program that converts a mere cloud of points into a web of polygons. Before these polygons are returned, there are a number of *black-box* systems in place to determine how to mesh the cloud.

For example, the user-defined points could simply be a polygonal model. In this case, GCore will merely connect the dots into the appropriate polygonal mesh. However, if the user has defined their object as a SubPatch object, it is GCore's job to appropriately mesh the cloud of points into its new smooth form by passing the traditional polygonal mesh through our sophisticated subdivision-surface meshing engine. Other *black boxes* currently in place are the spline curve solver and the metaballs-meshing engine. With the object-oriented nature of our GCore engine, NewTek will be able to add new forms of geometry engines in the future as new techniques for object creation arises.

ANIMATION PIPELINE

Once the object is prepared for animation, it is important to understand the order of actions inside of Layout. There are several steps that must be accomplished before objects are ready for rendering. With respect to geometry, these steps can include simple morphing, bone deformation, displacement (internal followed by plug-ins), and finally general transformation (move, rotate, size and stretch).

During the animation of these points, LightWave needs to mesh any SubPatch object. The order of this meshing is extremely important, as it will greatly effect the look of the final result. For example. Most users will want to use bones on a low-resolution mesh (pre-meshed) for something like a character, but a displacement on a water plane should be applied after the meshing to allow for higher resolution geometry for a finer result.

There is a similar ordering situation with Displacement plug-ins. The displacement handlers are applied to the geometry in the same order in which they appear in the list window—the handlers at the top of the list are applied first and the handlers at the bottom are applied last. This is very important to keep in mind as many of them are additive and thus will create different effects based on the order they are applied.

By keeping the transformation order in mind, you will have better results when applying different effects. It will also assist you when changing the subdivision order for greater control of your objects. There is a lot of technology in LightWave [6], but if you can understand the major components and how they interact you will be well on your way to mastering the application.

RENDER PIPELINE

The LightWave 3D render engine has always been considered the fasted, best looking, most flexible rendering engine on the market. With LightWave [6], added to that list is our most powerful and widest pipeline. The LightWave [6] render pipeline gets its power and accuracy from the 320-bit IEEE floating point render throughput. This means that each pixel calculated during render time has a potential for 320 bits of data. This data is in the form of color, alpha, z-buffer and many other render buffers that can be added to the final render.

The floating point nature of this data ensures extreme mathematical accuracy for gorgeous renders every frame. While the data is processed at a very high bit depth, there are a number of engines in the pipeline that can add specific effects or phenomena to the rendered image.

The LightWave [6] render pipeline consists of a fully distributive ray-tracing engine with shadows, reflection, refraction and translucency, three-dimensionally textured volumetric lights, cartoon ink and paint rendering, depth of field, adaptive sampling, motion blur, lens flaring, HyperVoxels rendering, radiosity, caustics and many other real world effects. By incorporating all of these technologies into one cohesive render engine, speed is enhanced and inter-operability is maintained. LightWave [6] is unique in having all of these functions integrated into a single pipeline. It is this holistic approach to rendering that has put LightWave ahead of the pack in previous revisions and with this incredible list of technologies LightWave [6] has reset the measure for 3D graphics render engines.

Along with the enhanced render pipeline and new render technologies, LightWave [6] now supports high dynamic range values. This allows numeric entry of many scene attributes to go well beyond the normal range of values. These increased values are then incorporated in during render calculations and used in calculations for effects such as radiosity and final compositing. Although a displayed pixel can never exceed 100% saturation, having pixels that contain values beyond 100% is essential for accurate secondary lighting with radiosity as well as for use in image processing and final composites.

VIPER

The VIPER (Versatile Interactive Preview Render) window provides the user with an interactive previewing system that can be used for full-scene evaluation and modification of textures, quick previewing of volumetric lights, HyperVoxels previews and can also be added to third-party extensions. The VIPER window traps much of the extra render data the LightWave [6] generates and leverages that data to allow instant updates to changes in a scene.

This window also allows surfaces to be selected by simply clicking on the object in the image window. VIPER can also be used as a quick animation preview generator. This is extremely useful in the case of volumetric and HyperVoxels effects since it can create an animation using only the current element. VIPER is a very powerful tool for quickly making adjustments to elements in a scene while avoiding costly test renders. It is also the bedrock for an expanding feature set for compositing with 3D data.

THE GRAPH EDITOR

The graph editor has been updated radically with greater functionality and streamlined workflow. Many changes have been made to the user interface as well as to the internal curve data format. The following is a quick hit list of those new features:

- Multiple curve types in a single curve
- New curve types, including bezier
- Multiple curve evaluation and modification of dissimilar items (view and edit any curve in a scene simultaneously)
- Create Favorites sets of curves for simple recall of editing environments
- Interactive cut and paste of key frames between any curves selected
- Bounding box or shift-click selection mode for multi-key editing
- Interactive proportional scaling of keys using cursor as center
- Data display over keyframe with mouse pointer
- Contextual pop-up menus in graph edit and graph storage windows
- Lock keys so they are uneditable
- Key bins for storing and inserting useful sets of key frames
- Velocity and speed curves in background views
- Copy and paste entire curves quickly
- New Footprints feature for creating a *back-up* of curves. Allows curves to be reset to their footprint position.

LIGHTWAVE [6] FOR EXPERIENCED LIGHTWAVE USERS

There are many new and wonderful things about the LightWave [6] system. At first glance, you will notice that this program still looks familiar to you—it still looks like LightWave. However, this application has undergone an incredible overhaul. The LightWave development team has made great strides towards adding amazing new technologies, streamlining workflow, while still maintaining an environment that will make LightWave 5.6 users comfortable. This section will help assist you in quickly traversing the new interface design.

Q: What happened to Get and Put?
Both environments, Layout and Modeler, can now share data through the Hub. The Hub is a central repository for data that removes the need for get and put options.

Q: How do I render? I can't find the Begin Rendering button?
This is now accessed from the menu. Simply choose **Actions** > **Render** > **Render Scene** or press F10.

Q: An object I created in Modeler loads into Layout with all the layers becoming separate objects. What happened?
Objects now consist of an unlimited number of layers, just like most paint packages. When you save an object, all of the layers are saved in the file. If you have some boolean cutting objects, etc. that you obviously don't want to be visible in Layout, use the Layers panel and make them invisible.

Q: How do I select between Object, Bone, Light and Camera modes?
In LightWave 5.6, there were two different places to choose your selection modes. In modeler you could pick between Point, Polygons and Volumes on the bottom of the interface. In Layout, you chose between Objects, Bones, Lights and Camera on the upper left side of the interface. In LightWave [6], you will find all selection options on the bottom left of the interface just as you do in Modeler. This allows more space on the toolbar attached to the side of the screen. Of course, item edit mode, in Layout, can also be changed with the hot keys O, B, L, C or by clicking directly on the item.

Q: There used to be buttons on the top of the screen. Now I only have three menu tabs. What gives?
In LightWave 5.6, there are a series of floating palettes we call panels. These panels were tied to buttons across the top of the interface. There were panels for Objects, Surfaces, Images, Lights, Camera, Effects, Options and Network. The top of LightWave[6] is now occupied by menu tabs rather than palette buttons. These menus are **Actions**, **Setting**, and **Extras**. The palettes have been divided into categories: Editors, Item Properties and Settings. The Objects, Bones, Lights and Camera Properties panels can be accessed by selecting **Item Properties** from the bottom of the screen, when Layout is in the appropriate edit mode. For example, selecting **Item Properties** when in the Camera edit mode will activate the Camera Properties panel. Changing edit modes will also change the panel context assuring perfect sync for the user. Pressing the P key just as the case in LightWave 5.6 can also access these panels.

There are actually two sets of menu tabs, one in Layout and one in Modeler. These menus replace the LightWave 5.6 mechanisms for activating panels in Layout. In fact, the menus in LightWave [6] function exactly as the buttons did in Modeler 5.6. The panel buttons utilized in LightWave 5.6 actually had duplicate meaning in that on one side, in Layout, they were used to activate panels and on the other, in Modeler, they were used to change the left-hand toolbar. The Modeler paradigm was true to the menu-tab design and so we have unified he two environments. You will find there are many similarities in the menu order. In particular, the first menus relate to item creation and modification while the later menus would be considered utility or options menus.

The Layout menu menus are fairly straight forward. The first menu, **Actions**, holds the controls for adding/removing items and modifying them. The second menu, **Settings**, offers access to different item settings for cameras, objects, lights and bones, as well as some scene level settings for the backdrop, volumetrics, compositing and image processing. The third and final menu in Layout, is labeled **Extras**. This menu contains many utility options for the program. These includes some visibility options, display settings, as well as preferences for customizing the interface and adding plug-in features.

Q: Where are my panels?
The floating palettes in LightWave 5.6 have been moved from the top of the screen into a series of editors and options panels. The editors include Scene Editor, Graph Editor, Image Editor and Surface Editor, while the options panels include Display Options and General Options, which can be accessed from the **Extras** menu or through the familiar shortcuts of D and O.

You will also note that all panels are accessed through blue buttons. In fact, the color coding for button types has remained the same from LightWave 5.6. While the LightWave 5.6 panels are now found as blue buttons on the three menus, there are other blue button options including Statistics, Edit Keys, Edit Menus, Motion Options and others.

Q: Where is Camera Target?
This can be found on the Motion Options panel at the top just under the **Parent** pop-up.

Q: Where is Align to Path?
Align to Path is now found on the Motion Options panel along with IK controls, parenting, camera target data and motion plug ins.

Q: How do I add Motion Plug-ins?
There are now two forms of motion plug-ins that can be added: Motion Handlers and Channel Handlers. For 5.6 style motion plug-ins, use the Motion Options panel on the IK and Controllers tab. Channel handlers are added through the Graph Editor on the Modifiers tab and can be applied to single channels of motion or any envelope in the scene.

Q: What is a Channel?
That is what you would probably call an envelope. Although, now it can also refer to a single channel of motion such as the X channel or the heading channel. By storing the data internally in the same format LightWave *perceives* motion channels and envelope channels as the same data and, thus, they can be linked and/or referenced to one another.

Q: What happened to my Item Panels: Camera, Objects, Lights?
All item properties panels that used to be accessed from buttons across the top of the screen can now be accessed by selecting the item type and pressing the **Item Properties** button at the bottom of the screen. You can also use the familiar key equivalent P.

Q: There is no IK Options button? How do I add IK to my items?
The IK options are now consolidated with other motion options on the cleverly labeled Motion Options panel. On this panel you will also find **Parenting**, **Camera Targeting**, and controls like **Align to Path**. IK and **Align to Path** are now assignable on a per-axis basis.

Q: What happened to the Parent button?
Parenting can now be found in two places. The standard **Parent** pop-up is found on the Motion Options panel. However, a simpler approach to parenting is to drag and drop the items on the Scene Editor panel. Simply drag the child item to the parent and drop it! *Ahhhhhh.* Oh, by the way, you can now parent any item to any other item in the scene! *Phew*!

Q: I notice I have to set my Segment Memory higher to reduce segments. What gives?
Memory is cheap! Seriously though, now LightWave [6] does all of it's internal render calculations in floating-point color space and with up to 320-bits per pixel! This enhanced pipeline allows for much better color accuracy and depth, but requires more memory.

Q: Hey, where is Depth of Field?
Still on the Camera Properties panel. To reduce the number of floating panels, we have reorganized certain things into tabs. **Field Rendering** can be found on the bottom of the Camera Properties panel under the Stereo and DOF tab.

Q: Where is my Effects panel?
All of the Effects options can be accessed from the **Settings** menu. You will see an Effects section on the tool bar and in that section are all the options: Backdrop, Volumetrics (used to be fog), Compositing and Image Process.

Q: Some of my surface attributes seem to be missing. Where are Additive, Sharp Terminator and Edge Transparency?
Some of these options have been upgraded to have percentages instead of simple toggles. With the new space required to add value fields we have moved Additive and Sharp Terminator to the Advanced tab on the Surfaces Editor. Edge Transparency is now a shader. Add it on the Shader tab.

Q: These floating palettes are cool, but sometimes I want to close them all at once. Could you add that feature?
Sure. Just a second...OK. There it is. I just added a keyboard equivalent to allow you to hide all panels with the TAB key. Try it out? Now that is customer service!

Q: Why is there a percentage value on the outside of bump map?
That is a global bump control so that after you have made a complex texture you can modulate it globally. Nice.

Q: The new surface by object is pretty cool. But sometimes I liked the old 5.6 method better. Can I make it so that surfacing works like it did in 5.6?
Are you sure? Well, OK. On the Surface panel, you can chose edit by scene or object. Edit by object allows the new method where each objects surface is separated. Edit by scene goes back to the more global surface edit mode of 5.6. It's easy, just flip the switch!

Q: Can I animate surface colors?
Yes. Use the **E** button next to color.

Q: How can I save an object from Layout?
Well it used to be on the Object panel, but now you can get to it quickly on the **Actions** menu from the **Save** pop-up menu. If **Save Object** is ghosted you do not have the object selected. Select it and then you can save or save transformed from that pop-up.

Q: How do I add a null?
That is done through the **Add** pop-up on the **Actions** menu.

Q: How do I delete an image from the scene?
Select the image on the Image Editor and press the DELETE key!

Q: I loaded an object into Modeler and it didn't load into the foreground layer. Plus, my existing object disappeared.
First understand that each object can consist of an unlimited number of layers and Modeler is now a multi-document application. Your existing object didn't go away, you can select it with the Current Object pop-up menu located just to the left of the layer selection buttons.

Q: What happened to Modeler's Make button? Now I can't even make a box!!!
All geometry creation tools are now interactive! You don't need a *Make* button. Once you turn off the tool you are finished creating or modifying the item.

Q: Since many tools are now interactive, how do I numerically create, say, a box without dragging out something first?
From the tool's Numeric panel, simply select **Activate** from the **Actions** pop-up menu. This activates the tool with the last-used settings.

ADDITIONAL RESOURCES

Listed below are some valuable sources of information about LightWave 3D available in the public, electronic media.

Internet Resources

A LightWave-specific newsgroup and mailing list are maintained on the Internet. Here you can find new users asking questions about using LightWave and experts answering them. Also, many topics related to computer animation are discussed in these groups, such as the performance of various accelerators, CPU speeds, animation recording devices, and many more. The newsgroup's Internet address is: comp.graphics.apps.lightwave.

Internet mailing list information can be found at www.tv3d.com.

NewTek World Wide Web Site

In addition to information about NewTek products, upgrades, and the latest releases of LightWave software, our World Wide Web site (www.newtek.com) has tutorials, LightWave images and animations, technical support FAQs, tech support e-mail link, and links to related sites.

The URL for the tech FAQ index is http://www.newtek.com/tech/faqs/mainindx.html.

Internet ftp site

NewTek maintains an ftp site (ftp.newtek.com) on the Internet. Here you can find objects, scene files, images, and other items of interest to LightWave users. You may download and freely use anything you find at the ftp site.

Technical Support

The best source for help with installing or configuring software or hardware is the retailer from whom you purchased your NewTek product. While we have made every effort to keep your software and hardware trouble-free and easy to use, you may occasionally need help right from the source. If you have problems with NewTek supplied hardware or LightWave doesn't seem to be functioning as it should, please contact technical support in one of the following ways:

- By email: tech@newtek.com
- By fax: (210) 370-8030
- By telephone: (210) 341-8444. Technicians are available to answer questions from 8:00 AM to 8:00 PM Central Time, Monday through Thursday, and Friday from 9:00 AM to 6:00 PM Central Time.

Please supply in your communication or have the following information handy when calling:

- You computer's operating system and version (e.g., Windows NT v4.0, etc.)
- The version of LightWave you are using
- The amount of RAM in your computer
- Any relevant specifics about your system (display card type, memory managers, accelerator type, etc.)
- Your product serial number

☞ **NOTE**

Your product must be registered before you can receive support.

WHAT'S ON THE CD ROM

LightWave 3D is installed from the included CD-ROM. In addition to the program and support files (including plug-ins) necessary for LightWave to run, it also has optional content, such as objects, scenes and images which you can install according to your interests and available hard drive space–you can delete these files later if you need to. Be aware, however, that some of the exercises and tutorials in the manuals require this optional material.

RUNNING THE PROGRAM

LightWave 3D is started the same way as any other program on the particular platform on which it's installed. Start LightWave (to start Layout) or Modeler by double-clicking their icon, or by clicking their icon and selecting Open from the menu. Each program can be started independently; however, you may also access them by clicking the appropriate button on their respective interfaces.

OPTIMIZING RAM USAGE

If your computer is accessing virtual memory frequently during rendering, you may find a substantial increase in performance by installing more RAM on your machine or reducing the amount of RAM LightWave needs to store the render information.

One of the best ways to minimize the need of RAM is to reduce the color of the images used for texture mapping effects. Except for those used as a Surface Color, texture maps only need to be 256-level gray-scale images.

Other ways to reduce RAM usage are to render your animation in multiple passes or decrease **Segment Memory Limit** on the Camera Properties panel.

USING LIGHTWAVE IN A MULTI-COMPUTER ENVIRONMENT

Modeler will automatically recognize and load objects in DXF, Wavefront, and 3D Studio formats. Modeler can also export objects in these formats. To use objects from other 3D programs, try to save the objects in one of these three formats.

You can transfer any objects or animations between LightWave on the different platforms.

Any scenes, objects, envelopes, etc., created with previous versions of LightWave 3D are usable under this version. However, once re-saved, they will not be compatible with previous versions. (Note: See also the object export features in the *Appendix*.)

chapter 2
Conventions

CHAPTER 2: CONVENTIONS

TYPOGRAPHIC CONVENTIONS

The following conventions will be used throughout this manual.

Directory Structure

Except for a few system files, all of the software will be installed in a subdirectory called NEWTEK on the drive you specified during the installation process. As such, unless otherwise specified, subdirectories referred to in this documentation are located in the main NEWTEK drawer. (For example, if the discussion refers to the Images drawer, the actual path might be C:\NEWTEK\IMAGES.)

Typefaces

ALL CAPS Computer keys, directories, device names (e.g., Enter, C:\NewTek\Objects, Control+P, etc.).

Bold Names of menus, commands, requesters, fields, buttons, etc. are set in bold type.

Keystroke Combinations

KEY1 + KEY2 Simultaneous keystrokes. Hold the first key and press the second key.

Mouse Operations

LMB Left mouse button

MMB Middle mouse button (if applicable)

RMB Right mouse button

Selecting Single clicking something with the LMB so that it becomes active or selected.

Deselecting Single clicking something with the LMB so that it becomes inactive or unselected.

Activate Selecting an option by clicking on its toggle button.

Deactivate Unselecting an option by clicking on its toggle button.

Clicking Placing your mouse pointer over something and then pressing a mouse button. This nearly always means the LMB.

Right-click Clicking something using your RMB.

Double-clicking Rapidly clicking something twice.

Dragging Selecting something with your mouse pointer and continuing to hold the mouse button down as you move your mouse. This nearly always means with the LMB.

Attenti-cons

✋ This symbol will be used to highlight a discussion that warns the user about something. You should pay special attention to text marked with this symbol.

☞ This symbol will be used to highlight a discussion that is particularly noteworthy.

☝ This symbol will be used to highlight tips and suggestions which are usually of a time-saving nature.

ABOUT THE MANUALS

Three are three manuals. The one you are reading now, *Introduction and Tutorials*, gives you an overview of LightWave, covers the functions that are common to both the animation and modeling aspects of LightWave and contains several tutorials. *Motion: Animate and Render* covers how to move your objects, set lighting and render options, and other information on how to create your animations. *Shape: Model, Surface, and Light* goes over how to build your 3D objects, apply surface textures, add lighting to your scene, and more.

LIGHTWAVE OVERVIEW

LightWave 3D is generally divided into two separate environments: LightWave Layout and LightWave Modeler. (It is a common convention to refer to the animation and rendering toolset as *LightWave*, or just *Layout*, and to refer to the object creation toolset as *Modeler*.)

With LightWave you are the producer, art director, cinematographer and director, all rolled into one. Your actors consist of objects that you build in Modeler or import from another program. You place lights to illuminate them, assign surface colors and values to *clothe* them, and then choreograph their movement to animate them. Once you have your actors in place, a camera records all of their movements, creating images of their movements that you can compile into animations.

LightWave can create animations and still images for virtually any use, from tiny 16 x 16 pixel Web graphics, all the way up to a resolution of 16,000 by 16,000 pixels, and anywhere in between. A multitude of preset resolutions are also provided including industry standard D1 and D2 (NTSC and PAL) video resolutions.

You create animations by generating a single frame at a time, then recording each frame to some type of playback device, like videotape or a hard-disk playback device, or even film. Frames can also be saved in computer animation format, like AVI.

KEY LIGHTWAVE TERMS & CONCEPTS

The following are some common terms and concepts used with LightWave and its manuals that you should familiarize yourself with:

Alpha matte/image — This generally refers to a image where the brightness of each pixel is used to cut or partially dissolve out another image. These are generally grayscale or black-and-white images, but the brightness values can also be extracted from a color image.

Animation Channel

Refers to the different position, rotation, and scaling settings an item can have in Layout. It can also refer to other *envelopeable* things like light intensity. See also *motion channel.*

Axis

Refers to the XYZ coordinates used as the basis for positioning things in LightWave's 3D space. It is somewhat like the concept of left/right, up/down, and near/far.

Bone

For any object, you can define a skeletal system composed of bones. By moving bones around, you can change the shape of an object.

Bounding Box

A six-sided box conforming to the outer dimensions of an object. Commonly used as a quickly drawn stand-in for a more complex object.

Camera

As in real life, a camera records events. Looking through LightWave's camera shows you the view as it will be generated.

Cattiwompus

Weird. Mixed up. Unusual. Distorted.

Channel

See *Animation Channel.*

Child

See *Parenting.*

Control Mesh

A cage of points used to shape subdivision surfaces.

Default unit

This is the unit of measure (e.g., meter, feet, etc.) that is assumed, usually when no unit of measure is entered with the numeric data. In Layout, it is determined by the setting on the General Options panel. In Modeler, the setting is on the Display Options panel.

EndoMorph

An object containing one or more morph maps.

Envelope

An envelope is a way of setting a particular value that usually changes over time using a graphical input mode.

Flatness

Flatness is used as a threshold in determining if a polygon is non-planar. A flatness of 0 percent means the polygon is absolutely flat.

Flatness is computed as percentage deviation from a triangle (the "ideal plane") formed from the first two and last vertices of a polygon. All of the other points are measured relative to this plane. The largest deviation is divided by the total size of the polygon to get a percentage which is the flatness value.

For example, if a polygon is 1 meter wide, a .5 percent flatness means that no point will be outside the ideal plane of the polygon by more than 5 millimeters. (1 x .005)

Frames

LightWave works with frames. A frame is one image out of many that define an animation. There are approximately 30 frames per second in NTSC video, 25 frames per second in PAL video, and 24 frames per second in film.

GCore

Short for Geometry Core, the *engine* that sits underneath all animation and modeling tools.

Geometry

This refers to the positional layout of points and polygons for an object.

High Dynamic Range Image

This is an image with a very large intensity difference between the brightest and darkest pixels. In typical 8/24-bit images, the maximum possible intensity range is 255 times brighter than the darkest grey pixel (with a value of 1). Natural scenes and images rendered with radiosity can have dynamic ranges from 10 to 10,000 times greater than this. Recording this information requires use of an image format with higher precision.

Hub, the

A *message board* that LightWave modules use to synchronize information.

HyperVoxel

Volumetric rendering effects.

Item

An item in Layout refers to an object, bone, light or camera.

Intelligentities

LightWave's object format.

Keyframe

(Also known as just a *key*.) A keyframe is a frame for which you define an animation channel(s) (e.g., position or rotation) for an item in Layout. Animations are composed of a beginning keyframe, an ending keyframe and usually some number of keyframes in between. See also *tween*.

Light

A light in LightWave is generally used just like a light in real life. Lights illuminate a scene and allow you to see the objects within it.

Modal/Non-modal

A modal panel must be closed before you can continue working with the rest of the application. A non-modal panel lets you shift the focus between it and another part of the application without having to close the panel—you can continue to work elsewhere in the current application while the panel is displayed. Modeler's Numeric panel is non-modal because you can do other things while it is open. In contrast, Modeler's Display Options panel is modal because you have to close it before you can continue working.

Motion Channel

Generally the same as *Animation Channel*, but only refers to position, rotation, and scale. (i.e., not light intensity.)

Non-planar

Refers generally to a polygon where all points do not reside in the same plane and can only occur with polygons using more than three points. Non-planar polygons can cause erratic rendering errors.

Normal

The imaginary line projecting out perpendicular to a surface at any point. A polygon surface normal is represented as dashed lines on selected polygons in Modeler. LightWave sees polygons or faces of an object only from the surface normal side. A single-sided polygon (like a piece of paper) with its normal facing away from the camera will be invisible to the camera from that viewpoint (unless the surface is using the **Double Sided** option).

Null Object

An object that contains no geometry and will not show up in a rendered image. Nulls are useful for tracking, grouping (parenting) and using with inverse kinematics functions. They are also commonly used to control plug-ins.

NURBS

Non-uniform Rational B-Splines.

Object

An object is composed of points and faces. Points connected together to form a polygon define a face. Faces joined together form an object.

Origin

The world Origin is the absolute center of the LightWave universe. A local Origin is the center of an object. Both are defined by the XYZ coordinates of 0, 0, 0.

Orthogonal

Usually refers to a display that only has two axes at right angles from each other in the same flat plane.

Parenting

This refers to LightWave's ability to set hierarchical associations between items in a Scene. Generally, the parent item will have some level of influence on its *child*, whether it be position, rotation, size, etc. The child can also be a parent to another item.

PAVLOV

Parameterized Animatable Values Linking Objects and Variables is a central repository for all animatable values in a scene.

Phong Shading

A computer process whereby faceted object faces are shaded and rendered with a smoothed over appearance. LightWave uses a Phong shading algorithm in its rendering calculations.

Pixel

A pixel is the smallest unit of measurement in an image and is used to describe the image's width and height. A single pixel in a LightWave generated image can contain any one of 16.7 million colors.

Plane

A two-dimensional (i.e., flat and level) surface. You might want to think of a plane like a piece of glass that was infinitely large, but had no depth.

Plug-in

A program that works with and extends the functionality of LightWave. Plug-ins are available for object replacements, object displacements, surfaces/shaders, motions, final image filtering, and more.

POV

Point of view

Radiosity

The reflection of light off of diffuse surfaces.

Ray Tracing

A process whereby a "ray" from a light source is followed as it interacts with objects in a scene and travels into the camera. Ray tracing allows for such realistic events as shadows, refraction, and reflection.

Render

Rendering is the computer's process of calculating and generating an image based on the values you have selected for the different options in LightWave.

Scene

A LightWave project defining the objects loaded and their motions, the number of lights and their values/motions, the resolution of the final image, special effects, camera settings, etc. This ASCII text file is generally saved from Layout.

Scrub

Usually refers to the action of dragging a slider and seeing/listening to its effect on video/audio.

Session

A single use of an application. A session starts when you first boot the application and ends when you exit.

Shaded mode

Generally refers to a viewport which has its Rendering Style (Display Options panel or viewport titlebar) set to something other than wireframe. These show polygon surfaces with some level of shading.

Spline (Curves)

LightWave uses splines or curved paths between keys while moving items about. When modeling, splines refer to open or closed curves.

Spline Cage

Usually a three-dimensional object made up of connected spline curves.

Spline Patching

Adding polygons to fill in areas outlined by splines.

Subdivided

Generally refers to the result of increasing the number of polygons by dividing existing ones. The overall shape of the subdivided polygons may or may not change depending on the method of subdivision.

SubPatch

Modeling mode where polygons become a cage controlling an underlying mesh of NURBS.

Surface

Essentially, this is the skin of an object. A single object can have multiple surface names, each with its own independent attributes (e.g., color), and multiple objects can share the same surface name(s).

Tangent

A straight line that makes contact with a single point along a curve.

Texture

A texture is an attribute of a defined surface, or an entire object, in the case of displacement and clip maps.

Tween

The internal process of calculating the animation channel values between keys.

Vertex

The point at which the sides of a polygon intersect.

V (Vertex) Maps — Abbreviation for vertex maps. This is additional information associated with object points (vertexes), like weight, UV and morph maps.

VIPER — Versatile Interactive Preview Render window which provides the user with an interactive previewing system.

Volumetric Lights — Special light property that allows you to see the beam of light.

Weights — See *V Maps*.

Wacky — Crazy. Zany. Mixed-up. Cool.

WORKING WITH THE INTERFACE

In no time, you'll find yourself mastering LightWave's intuitive interface. LightWave uses very few picture-icons that may confuse you. Instead, most functions are listed on the interface panels in plain text. There are a few other conventions involved with the LightWave interface which are listed below:

Button — Refers to an area on the screen that you click on with your mouse to cause some function to occur. Generally, only a single click is required. There are also special types, like toggle, pop-up menu, and envelope buttons. Some buttons become highlighted, indicating a chosen or active status.

Contextual Pop-ups — Refers to *context-sensitive* pop-up menus. Such menus appear when you Shift+Ctrl+LMB and are aware of the area the mouse pointer is over.

Dialog — See *Requester*

Drag Button — This a type of button is similar in effect to a slider, except the button does not move.. To use, just click on it and hold the mouse button, then drag the mouse. Depending on the parameter being adjusted, the direction and mouse button used will have different specific effects.

Envelope Button — A small button marked with an E. Selecting this button will display the Graph Editor where you can create an envelope for the setting. A highlighted envelope button indicates a value has an envelope applied. To turn off (i.e., remove) an envelope, hold the Shift key and click on the Envelope button.

Ghosted Item — LightWave will *ghost out* parameters that are not available to you. This is usually the result of a certain option not being activated. Selecting a ghosted item will display a message informing you why it cannot be used.

Information Field/Display — These are text displays found throughout the different panels. These displays cannot be changed directly and simply provide information and feedback.

Input Field

These are areas on the screen where you can enter data. In Layout, by clicking the pointer just in front of an existing value (a space should appear between your cursor and the first number), you can type in a new value and press Enter without having to delete the old value. Likewise, for those systems that support it, you can double-click on an existing value to highlight it, then type in a new value to replace it.

☞ ***NOTE***

Most panels with multiple input fields will progressively move your cursor from field to field after you press Enter or the Tab key. Pressing either key without entering any new data will leave the old value intact. In most requesters, once all values are entered, a final press of Enter will close it, saving you from clicking OK.

Mini-Slider

A button with two arrows on it, pointing right and left. Clicking on one and holding the LMB, then dragging to the right or left will raise or lower the value of the parameter next to it. In many cases, however, the slider does not encompass the entire spectrum of possible values.

Panels

Any one of the windows that open when you click on a button in LightWave. Many panels have additional tabbed sections that are selected when you click on a tab.

Pop-up Menu

These buttons have a downward facing arrow on their right edge. To use, click on it and hold the LMB. The menu will pop up and as you move your pointer over the menu, each item will become highlighted. When the desired selection is highlighted, release the LMB.

If you decide not to select an item, simply move the pointer off the menu and release the LMB.

Certain pop-up menus contain lists of objects, images, and lights, which are normally listed in the order they were loaded or created.

Requester

This is also known as a dialog box. These appear on the screen for operations like file loading and saving. This also refers to smaller windows that appear requesting the user input data into various fields.

Reset Area

A non-active open areas (e.g., not a button) on the Modeler toolbar which acts as reset button, much like you might use the Esc key on other applications.

Scrollbar

See Slider

Shaded Display

See *Solid-Shaded Display.*

Slider

This type of slider allows you to modify a setting by dragging the slider's button along the bar. Alternatively, you can click to the right or left of the button or use the arrow buttons at either end to incrementally change the setting value. (Also known as a scrollbar.)

Solid Shaded Display This refers to a non-wireframe display mode where some level of surface texture detail is visible.

Texture Button A small button marked with a T. Selecting this button will present a texture panel allowing you to define a texture for the chosen parameter. A highlighted texture button indicates a texture in use for the given parameter. To turn off (i.e., remove) a texture, hold the Shift key and click on the Texture button.

Toggle Button A small button that when clicked on will becomes highlighted with a check mark inside. This indicates the feature listed next to it is active.

BUTTON CONVENTIONS

Buttons are generally color-tinted to indicate their operation.

Cyan Brings up a dialog or panel. Often, the button will become highlighted while the dialog/panel is open. Clicking the button again will close the dialog/panel and accept any changes. The button will also unhighlight if the dialog/panel is otherwise closed.

Magenta Commands or processes that act immediately

Green Pop-up menus

Yellow Tools. These will be highlighted while the tool is in use. If the tool is in a pop-up menu, a checkmark will indicate it is active. Selecting it again will deactivate it.

Buttons will appear ghosted when the context of the data and selection make them obviously inapplicable.

chapter 3
Common Interface Items

CHAPTER 3: COMMON INTERFACE ITEMS

This chapter covers common interface items as well as panels and dialogs that are shared between modules, panels, plug-ins, etc.

PLUG-INS

Plug-ins can greatly increase the available features. Some plug-ins are tools. As such, they can be active or inactive just like a standard tool. If they are active, there will be a check mark next to them, if the plug-in appears in a menu group list. If not, they can appear as a button on the toolbar. In this case, it will appear highlighted when active. (See the Introduction manual for information on how to customize menus.)

All of the plug-ins that come with LightWave should already be installed and available. However, there may be times when you need to re-add one because of an update or just add a new third-party plug-in.

Generally, plug-in files will be stored in various subdirectories in the LightWave PLUGINS subdirectory. Some plug-ins work only in Layout or Modeler, while others work in both. You can multi-select plug-in files when adding—if supported by your OS—, and attempting to add a plug-in that is not meant for that module (i.e., Layout or Modeler) or adding a plug-in that is already added will not do any harm.

To add plug-ins to Layout:
Choose **Extras** > **Add Plug-ins** and select the plug-in file from the file(s) dialog that appears.

To add plug-ins to Modeler:
Choose **Objects** > **Preferences** > **Add Plug-ins** and select the plug-in file(s) from the file dialog that appears.

If the plug-in is not assigned to a button, it will appear in the **Objects** > **Additional** pop-up menu. You can then move it or leave it, as desired. Menu editing is discussed later in the chapter.

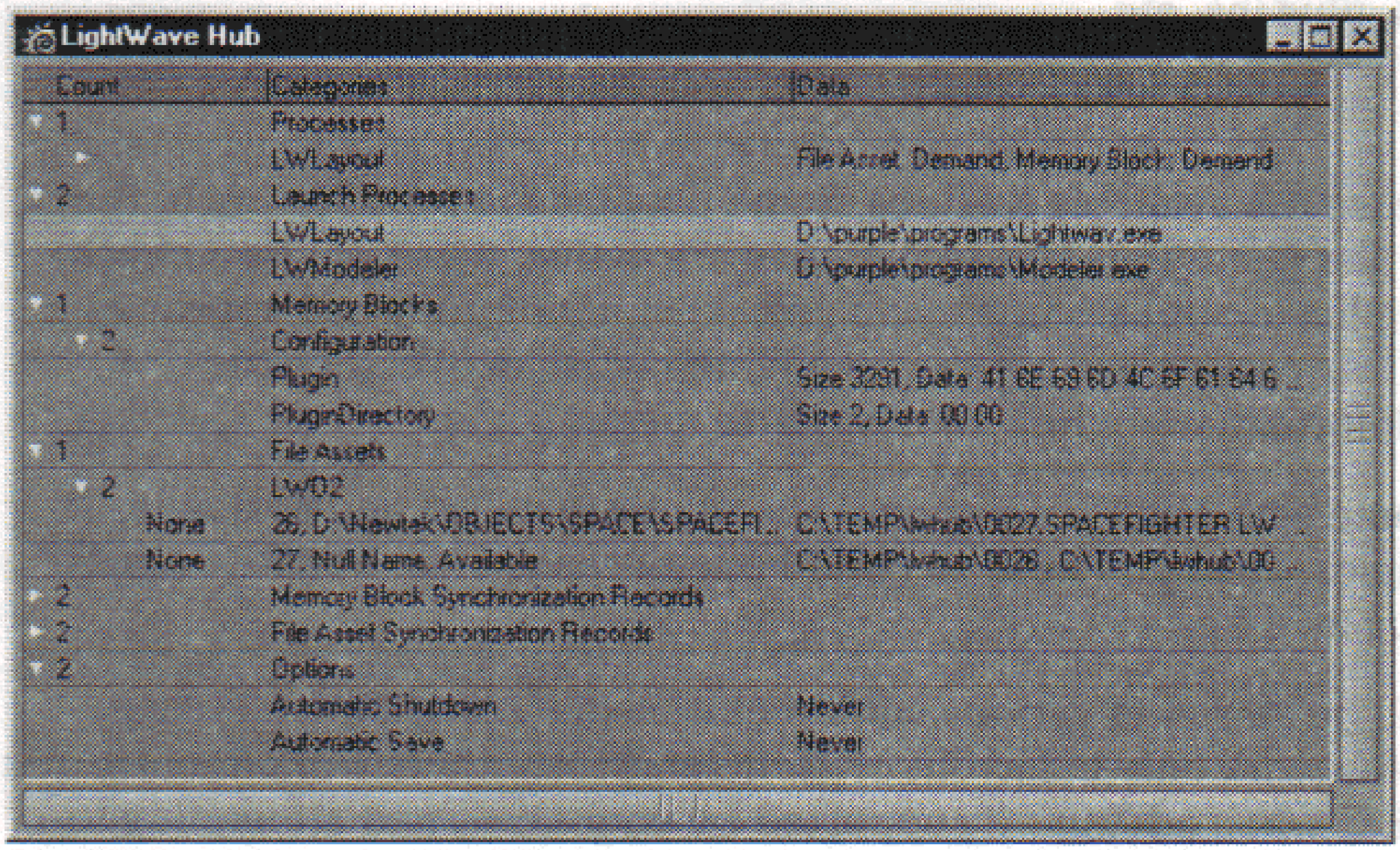
☞ ***NOTE***
A single plug-in file can have many functions, some internal and not directly accessible by the user. Thus, when you add one, it may report back that it has added more than one plug-in. This is normal.

THE HUB

The Hub is essentially a *message board* that LightWave modules use to synchronize information. It contains things like *synced* object filenames and configuration memory blocks. When the same object is loaded into both Layout and Modeler, changes made to the object are automatically synchronized. If the object appears in Layout, but not Modeler, you can quickly load it into Modeler by selecting it from the object pop-up menu—initially, the name will be ghosted.

☞ ***NOTE***
If you have modified an object in Modeler (without saving) and then load the object file into Layout, the modified version will appear—since it is synced—not the data from the object file.

The Hub is run automatically when you run LightWave. Basically, it is a background process and you shouldn't need to directly interact with it. However, you can bring up its interface by clicking its icon (when running).



LightWave Hub

Count			Categories	Data
▾ 1			Processes	
	▸		LWLayout	File Asset: Demand, Memory Block: Demand
▾ 2			Launch Processes	
			LWLayout	D:\purple\programs\Lightwav.exe
			LWModeler	D:\purple\programs\Modeler.exe
▾ 1			Memory Blocks	
	▾ 2		Configuration	
			Plugin	Size 3291, Data: 41 6E 69 6D 4C 6F 61 64 6 ...
			PluginDirectory	Size 2, Data: 00 00
▾ 1			File Assets	
	▾ 2		LWO2	
		None	26, D:\Newtek\OBJECTS\SPACE\SPACEFI...	C:\TEMP\lwhub\0027.SPACEFIGHTER.LW ...
		None	27, Null Name, Available	C:\TEMP\lwhub\0026 , C:\TEMP\lwhub\00 ...
▸ 2			Memory Block Synchronization Records	
▸ 2			File Asset Synchronization Records	
▾ 2			Options	
			Automatic Shutdown	Never
			Automatic Save	Never

If you expand the Launch Processes menu, you should see applications that the HUB is aware of (e.g., LWLayout, LWModeler). You can launch the application by double-clicking it. If you single-click (to select it) and then press the L key, you will also launch the application. If you press the DELETE key instead, you will delete the application from the launch list. (This has no effect on the program files you your hard drive.)

☞ **NOTE**
The Launch Processes menu will be empty initially. However, once you run Layout or Modeler entries will be added.

Under Windows, the Hub icon will appear in the System tray. You can right-click on it to access a small menu:

Open	Open Hub window
Close	Closes Hub window
Launch	Launch applications that the Hub knows about
Properties	Lets you set options to quit the Hub after you exit LightWave
Exit	Closes the Hub program

Properties

You can set the **Automatic Shutdown** to various time intervals, after which the Hub will shutdown when there is no activity.

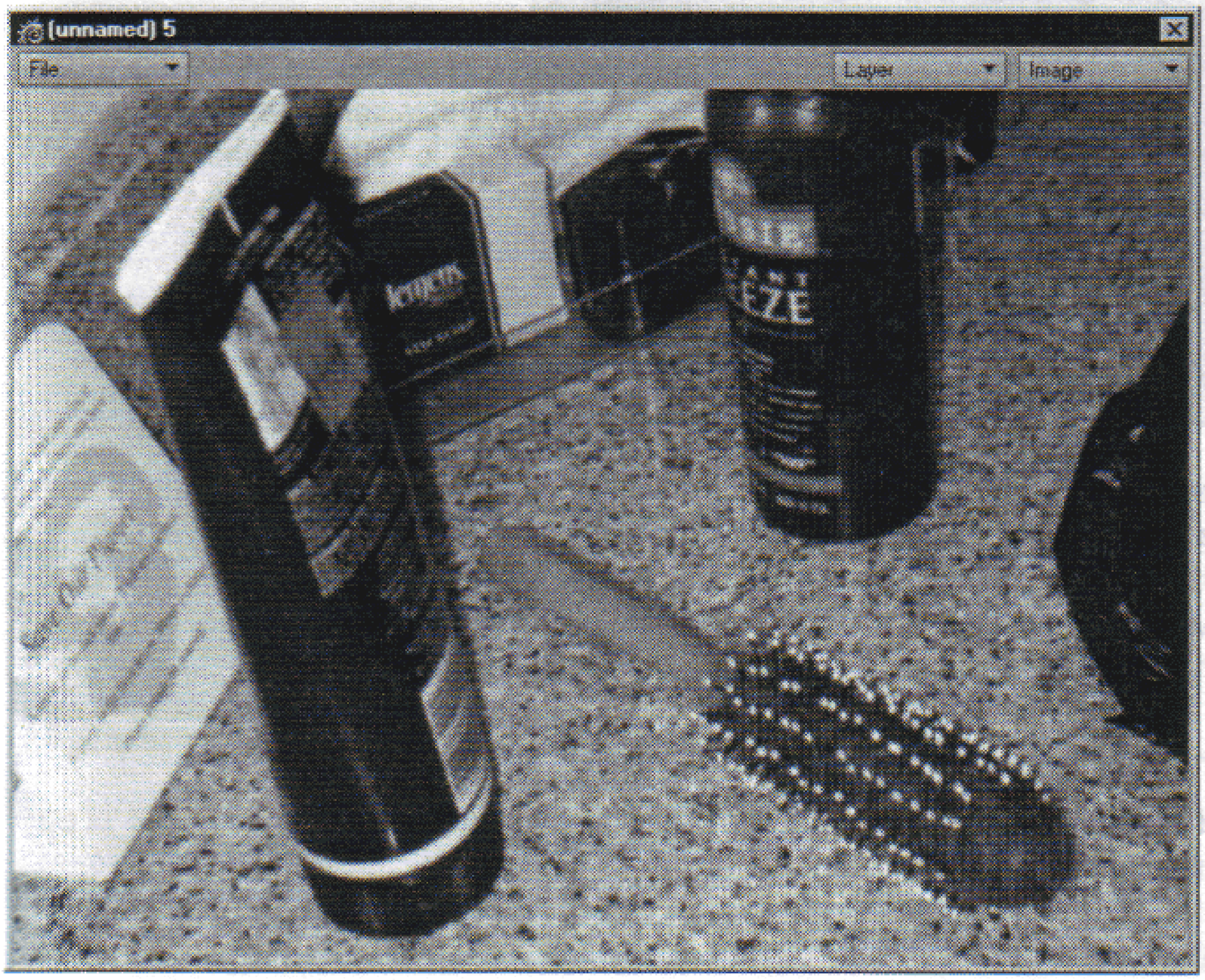
When Layout or Modeler are running, the hub will periodically request that the applications write their recent changes to temporary files. The frequency of this **Automatic Save** is set by the hub option. The temporary files are located in a LWHUB directory inside your normal system temporary directory. The files have names that include the base name of the object with a numeric prefix. So, if you are editing an object called MYOBJECT.LWO and you crash after an editing session, you can still find an *auto-saved* version of your edits in the temporary directory with a name like 0012.MYOBJECT.LWO.

☞ ***NOTE***

If you would like to run Layout and/or Modeler independently, append -0 to the command (e.g., Lightwav.exe -0. That's a zero, by the way.) You will have to load and save objects manually.

IMAGE VIEWER

The **Image Viewer** is used throughout LightWave (Render Display, Image Editor, etc.) to show an image using colors up to the capabilities of your computer's display. The image may be scaled if it will not fit entirely on your display.



From Render Display (Render Options panel)

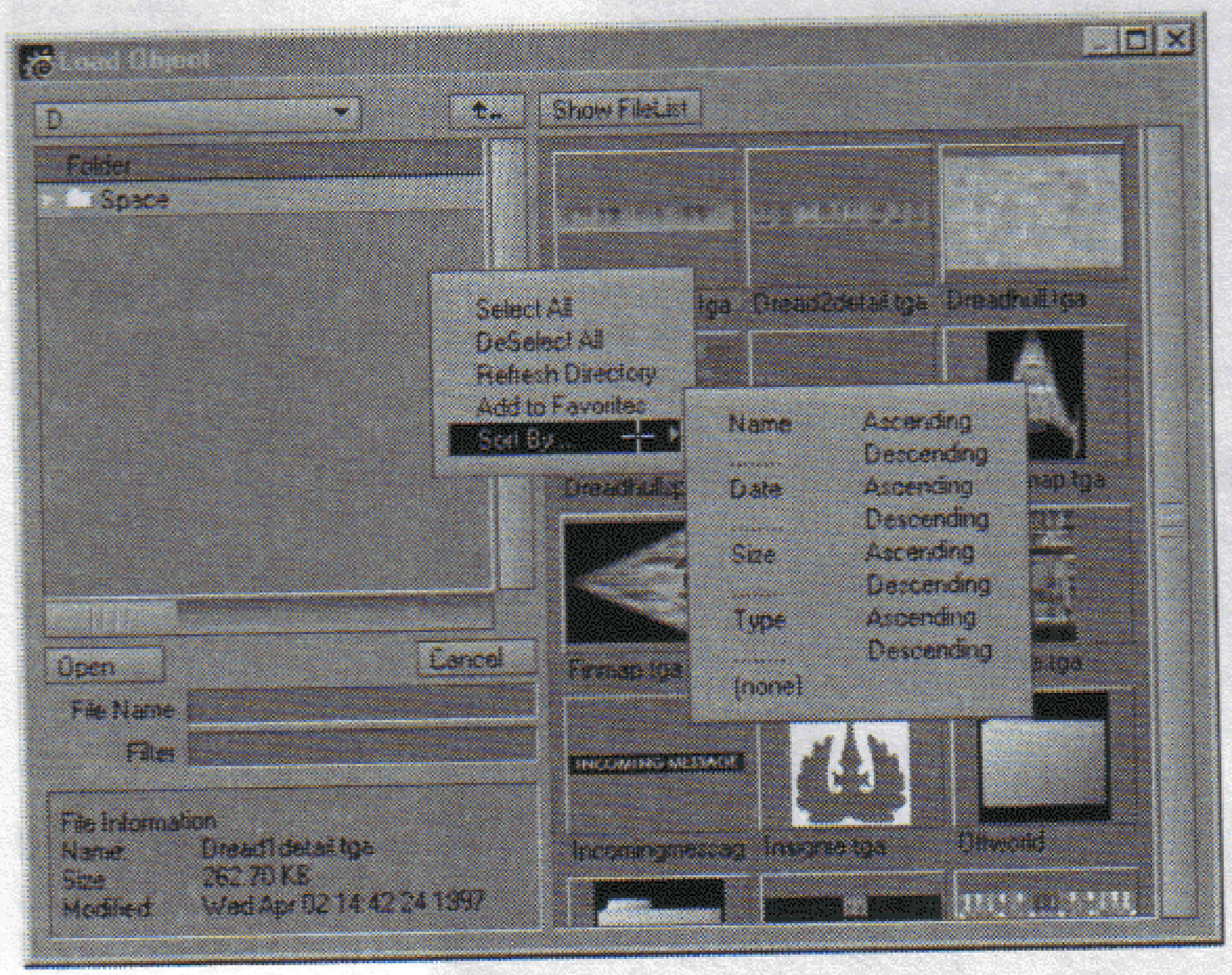
Using the **File** pop-up menu, you can save the current image to a file. Once you select the format type a file dialog will appear. Make sure you add the appropriate filename extension, if appropriate (e.g., WonderfulPic.tga). You can choose to see the regular image or the alpha image, if applicable, using the pop-up menu in the upper-right corner.

Once open, you do not have to close the Image Viewer window. In fact, if you do, any unsaved images may be lost. You can redisplay the Image Viewer—if it has been used during the current session—by selecting **Render Review** from the **Generics** pop-up menu on the General Options panel.

The Image Viewer can sometimes hold multiple images, depending on how it is being used by LightWave. In such a case, you can select them using the **Layer** pop-up menu. You can also clear layers from memory using the **File** pop-up menu.

VISUAL BROWSER

You can make the LightWave visual browser the default file dialog by choosing **VBFileRequester** on the **File Dialog** pop-up menu on the Interface tab of Modeler's Display Options panel.



File Information
Name: Dread1detail.tga
Size: 262.70 KB
Modified: Wed Apr 02 14:42:24 1997

Most of its features should be self-explanatory. The button above the list window will toggle between **Show Icons** and **Show Filelist**, allowing you to switch between an icon or list view. Right click on the list window or an item in the directory window to display the options pop-up menu. The **Add to Favorites** option will add the current directory to the drive select pop-up button in the top-left corner.

VIPER: THE INTERACTIVE PREVIEW WINDOW

Most people think of rendering engines as the part of the program that converts the scene file data into two-dimensional pixels of varying amounts of red, green, and blue that create a color image. However, some of these pixels can also have an alpha value, which will determine the transparency of each pixel.

In LightWave [6], this is only a small fraction of the data that is generated during a render. LightWave [6] can also generate z-buffer (depth), luminosity, diffuse, specular, mirror, shading, shadow, geometry, object, diffuse shading, specular shading and even custom surface buffers. At first glance, it may not be readily apparent why this is an advantage, however, you will see there are many advantages to having access to this extra information.

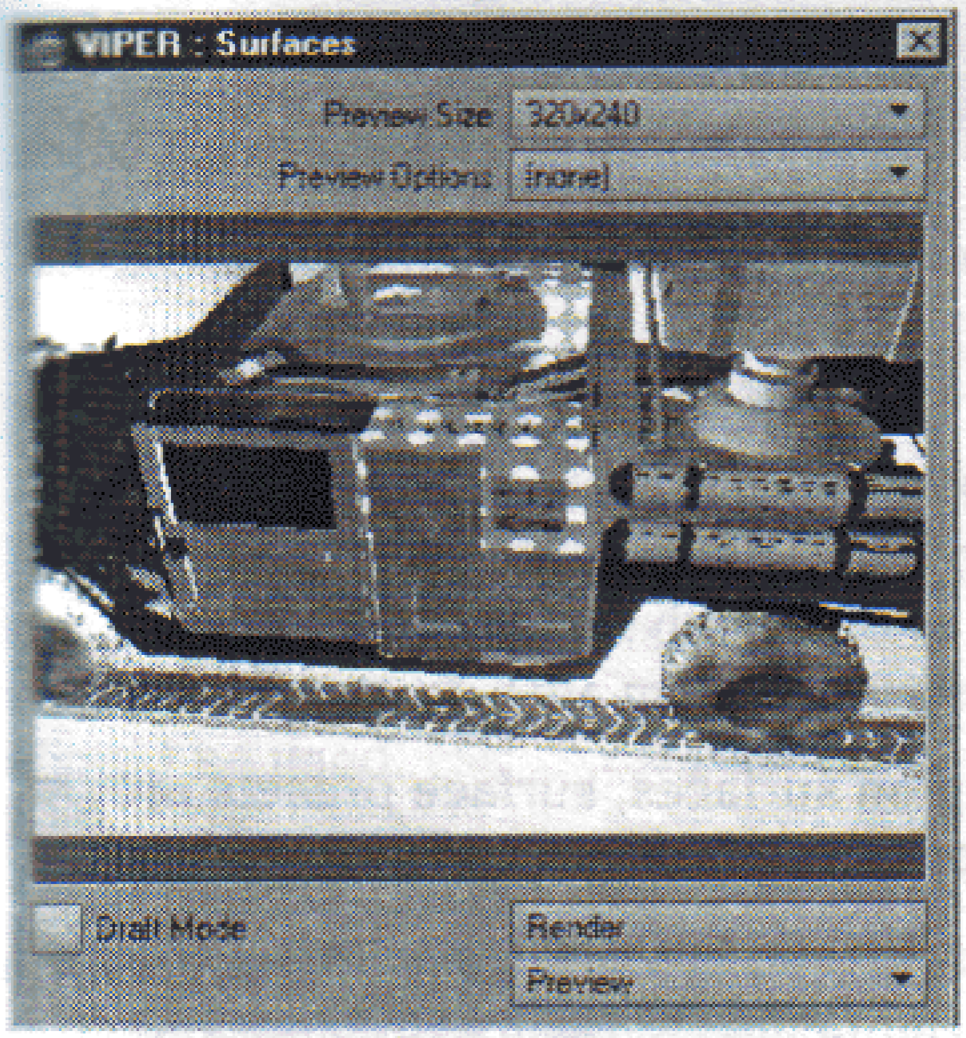
The *VIPER* system (Versatile Interactive Preview Render) allows users to make changes to what are typically render-intensive elements in the scene and get nearly instantaneous feedback. While many applications rely on Open GL for fast, but inaccurate, previews of a scene and others require a full scene render, LightWave uses VIPER to allow users to see changes of surface attributes, such as procedural textures, image maps, transparencies, refractions, bump maps, as well as volumetric light changes, general lighting changes, volumetric light texturing, HyperVoxels settings and even third-party party plug-in extensions.

The window uses the last rendered image data for the display. As you make scene changes, you will see those changes reflected. If you click on the VIPER image, the visible surface will become the current surface in the Surface Editor.

To accomplish this small miracle, VIPER displays a 2-D image preview window but also keeps all of the extra information stored at each pixel location. This way VIPER not only knows what color a pixel is, it also knows how far back on the z axis that pixel is, as well as what surface it relates to and many other important pieces of data.

Because VIPER has all of that extra data stored, it can change a surface color or specularity setting and show the result amongst the rest of the scene without requiring another full frame render. This will even show the appropriate shading, actual scene lighting changes, as well as backdrop color changes.

NOTE
To make VIPER accessible, you must activate the **Enable VIPER** option on the Render Options panel. When rendering, you should disable this option, since it increases rendering time and memory consumption.


VIPER : Surfaces

Preview Size: 320x240

Preview Options: (none)

□ Draft Mode

The **Draft Mode** option will use a lower-resolution for faster updates. Click the **Render** button to force a refresh. You can press the Esc key to abort the render. The **Preview Options** pop-up menu can be used to access display options—these will vary depending on what feature is using the window. You can select from various window sizes using the **Preview Size** pop-up menu.

For certain features, like surfaces, volumetric lights and HyperVoxels, double-clicking the image will add it to the Preset Shelf, discussed next.

The **Preview** pop-up menu operates in a manner similar to its cousin on the main interface. You can use this function to preview things like animated procedural textures, volumetric light textures, HyperVoxels, etc. Note that objects will not move like they would with a standard preview animation and pressing the Esc key will abort the preview creation process. (See the *Keyframing* chapter in the Motion manual for more information.)

☞ **NOTE**

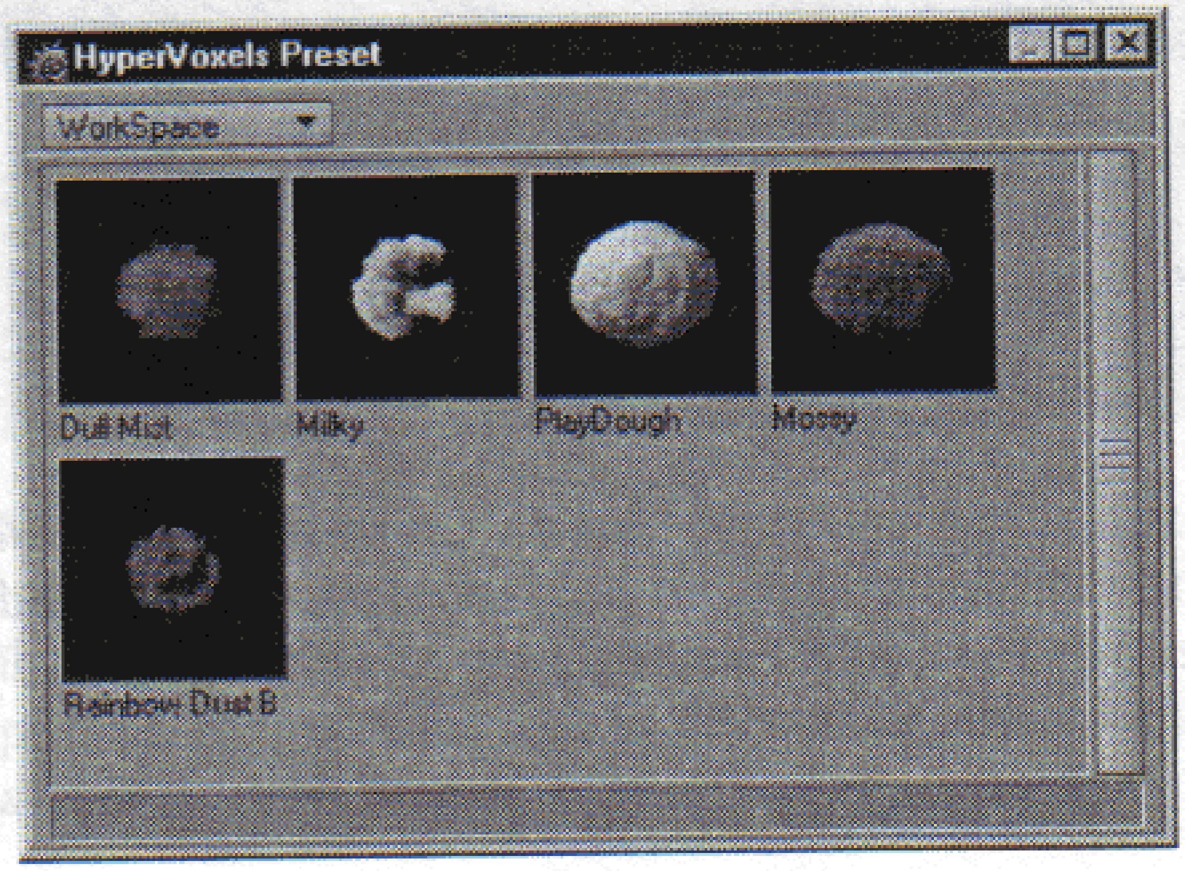
Since VIPER does not do a full-scene evaluation, there are some things that are not accounted for, like textures using UV coordinates or weight maps, ray-traced effects (reflections, refractions, shadows), shadow maps, fog, double-sided polygons, radiosity, light falloff, etc. As such, it is not a replacement for the render preview window.

☞ **NOTE**

To make VIPER accessible, you must activate the Enable VIPER option on the Render Options panel.

PRESET SHELF

The Preset Shelf is a sizable floating window that holds a list of previews along will all of the associated settings. It can be used for settings with surfaces, volumetric lights, HyperVoxels, etc. You add to the shelf by double-clicking on the preview window in the editor (e.g., Surface Editor) you are using or by clicking on the VIPER window. The shelf survives from session to session.



The shelf can be resized by dragging an edge of the window

The window is context sensitive, so if you are working on surfaces, surface presets are displayed, if you are working on HyperVoxels, those presets are shown, etc. etc.

Each editor has a default *library* named **Workspace**—a library is a grouping of presets for a particular editor. You can create a custom library by right-clicking over a preview and choosing **Library** > **Create**. For example, using the Surface Editor, you might make a library of *Wood* presets, a library of *Stone*, etc. The **Library** > **Empty Current** menu clears all of the presets from the displayed library.

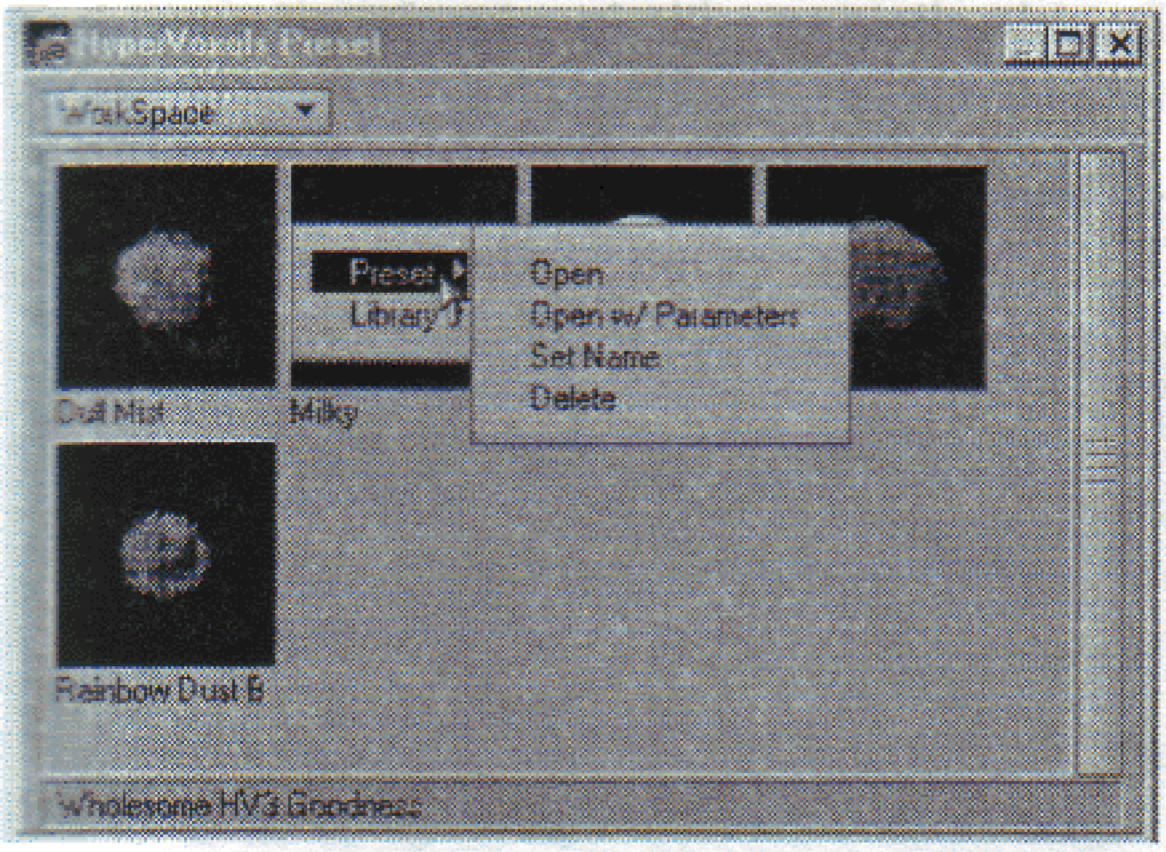
To apply one of the presets:

1 Double-click on a preview or select **Preset** > **Open** from the RMB pop-up menu.

2 Click the **Yes** button on the confirmation dialog.

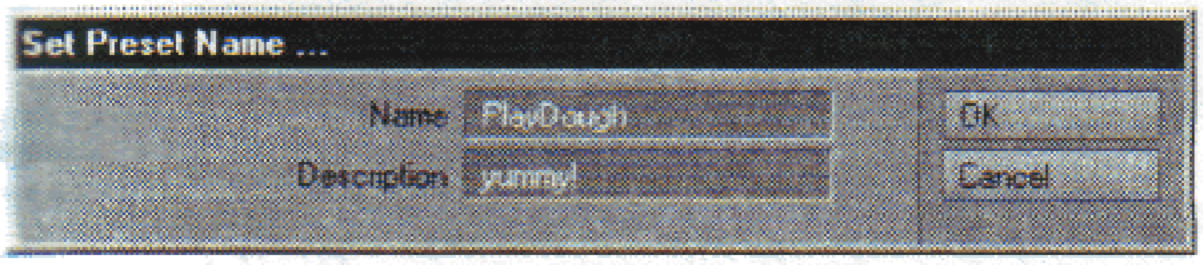
☞ **NOTE**

Preset > Open w/Parameters works with certain preset groups, like HyperVoxels, and allows you to pick which part of the settings you want to use.



Wholesome HVG Goodness

Choose **Set Name** to open a dialog box where you can set a **Name** to be used for the sample, it appears underneath the sample image. You can add a longer string of text in the **Description** field, which appears at the bottom of the panel when your mousepointer is over a preview.


Set Preset Name ...

Name: PlayDough

Description: yummy!

The **Delete** option removes the sample from the shelf.

KEYBOARD SHORTCUTS

Keyboard shortcuts are displayed on the right-side of buttons, if applicable. Keystrokes requiring the use of the SHIFT key are displayed in uppercase if they are letters. (e.g., A) Unshifted keystrokes are displayed in lowercase. (e.g., a) For example, M is SHIFT + M and & would be SHIFT + 7. This manual, however, always explicitly indicates when the SHIFT key should be used.

WARNING

You need to be aware of the state of your CAPS LOCK. If this is active, it have an effect on upper and lowercase keyboard shortcuts.

Panel-Specific Shortcuts

Some panels have their own special keyboard shortcuts. To use them, you must have that panel active. (Click on it, if it isn't already active.) If the panel is not active and that shortcut has another meaning for LightWave in general, that function will be run. Moreover, if the panel is active, but doesn't use the keyboard shortcut, it will be passed to the main interface.

LOADING AND SAVING FILES

LightWave generally uses the default load/save dialogs for the particular computer platform you are running (these are open to third-party support). These dialogs, by default, start looking in a specific directory which can be modified by editing LightWave's configuration file. (See the *Appendix* for more information). Double-clicking a file in a load dialog will load the file, while double-clicking in a save dialog will save over the filename you select.

All keyframe, object name, option, control panel settings, etc. are saved in your LightWave Layout scene file. Loading this file later will restore all of the saved settings.

FILENAMES AND EXTENSIONS

It is a good idea not to use spaces in hard drive, directories and file names for Scene, image, and object files. This can cause problems later if you use LightWave's distributed rendering feature.

LightWave uses several filename extensions for the different types of files it uses. Here are some of them:

.env	Envelope data
.lwo	LightWave object
.lws	LightWave scene file
.mot	Motion data
.p	Plug-in
.srf	Surface attribute file

Generally, LightWave will automatically add the appropriate extension to filenames when saving files, if one is not provided. However, this is not the case with most plug-ins.

Additionally, there are numerous standard filename extensions (.bmp, .iff, .tga, .wav, etc.) that LightWave uses, but are not specific to this application.

LIGHTWAVE PANELS AND DIALOGS

LightWave's panels and requesters are generally non-modal and can be left open even while you interact, say, with the Layout window and its controls.

Enter/Tab Keys with Input Fields

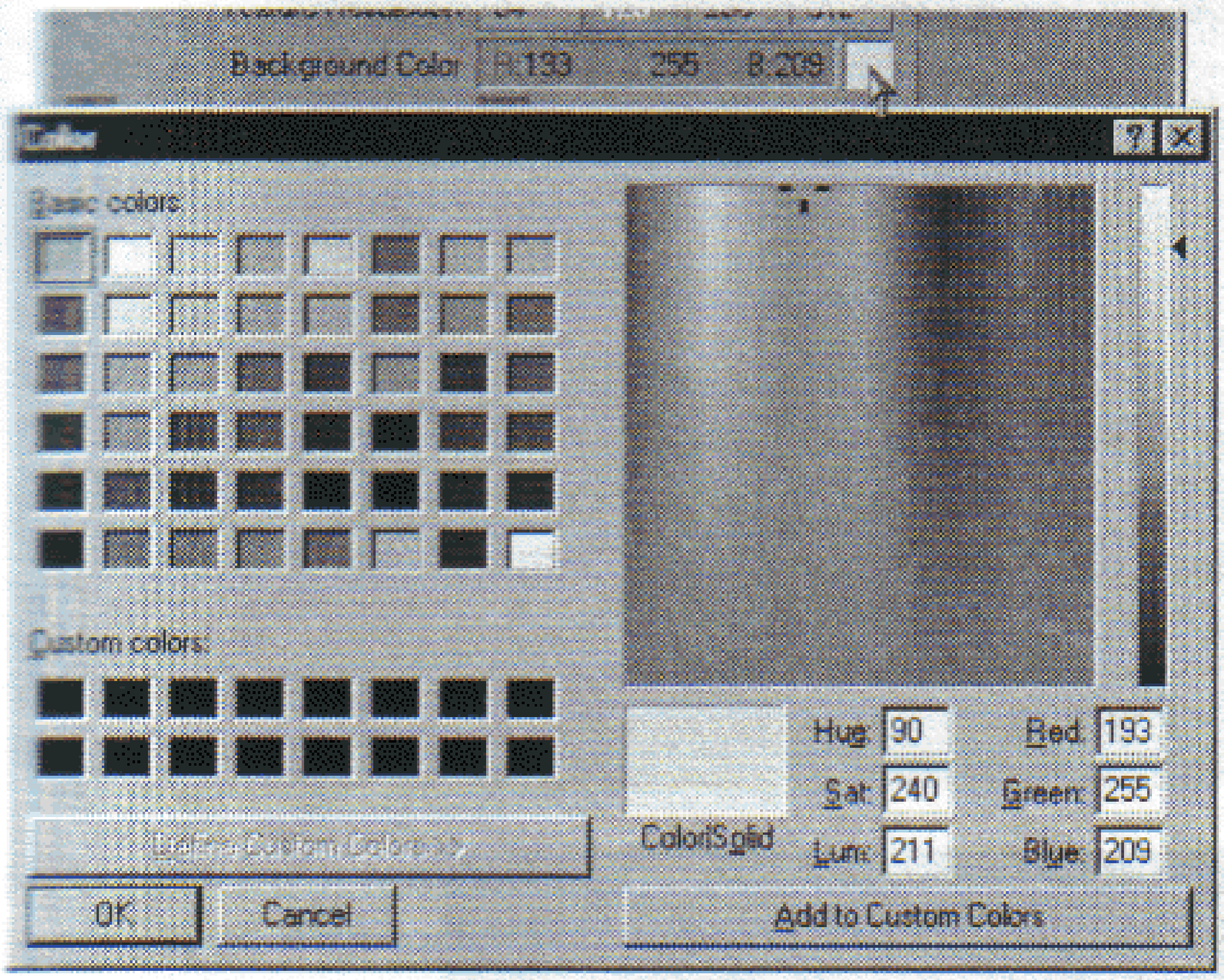
When entering values into dialogs that have multiple input fields, the Tab and Enter keys have special functions to save you time. Pressing the Tab key stores the entered/current value and automatically advances you to the next input field. The Enter key also stores value, but will dismisses dialog, if appropriate.

Yes and No

Whenever a requester is open that asks you a question and you have the choices of **Yes** or **No**, pressing the Enter key is the same as selecting Yes, while pressing the Esc key is the same as selecting No. Likewise, an Error, Warning or Reminder requester with a choice of **Continue** or **Cancel** can use Esc for Cancel and Enter for Continue. For those requesters with only a **Continue**, pressing either Esc or Enter will close the panel.

Color Selection

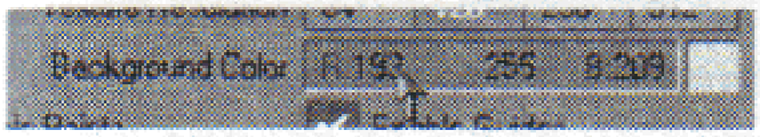
Where it is necessary to select a certain color (e.g., surface color), there will be a three-number color component display and a color swatch box. Clicking the color swatch box will open the color selection requester standard for your specific platform.



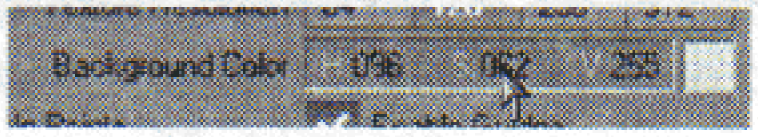
Background Color R:193 255 B:209

Hue: 90
Sat: 240
Lum: 211
Red: 193
Green: 255
Blue: 209

You can also drag your mouse on each color component to alter its value. By default, the standard **RGB** color space is displayed; however, if you right-click on the display, you may also use the Hue, Saturation, and Value (**HSV**) color space, if that is more familiar to you. HSV is based on the artist concepts of tint, shade, and tone. There are 16.7 million possible color combinations.



Background Color R:193 255 B:209



Background Color 096 062 255

☞ **NOTE**

Unlike **RGB**, **HSV** color components are not independent. For example, if Value (i.e., brightness is 0), you will not be able to change Hue (color) nor Saturation. Similarly, if Saturation is 0, you will not be able to change Hue. Moreover, reducing one component may cause another component to decrease.

☞ **NOTE**

LightWave will attempt to display the selected color to the best of your display's capabilities. This will generally result in an approximation of the color if you are using less than a 24-bit display mode.

Standard List Windows

Windows that contain lists of items (surfaces, scene items, menu items, etc.) all have a slider bar at their left to scroll through the list. If the list is hierarchical in nature, there will be arrowheads to the left of item names. If the arrowhead is facing to the right, then they are subordinate items to this *parent* item. Clicking the arrowhead will make it face down and also reveal the subordinate items.

If selection is appropriate, you can click on an item to select it. Ctrl + Click will select non-contiguous items and Shift + Click will select a range of items.

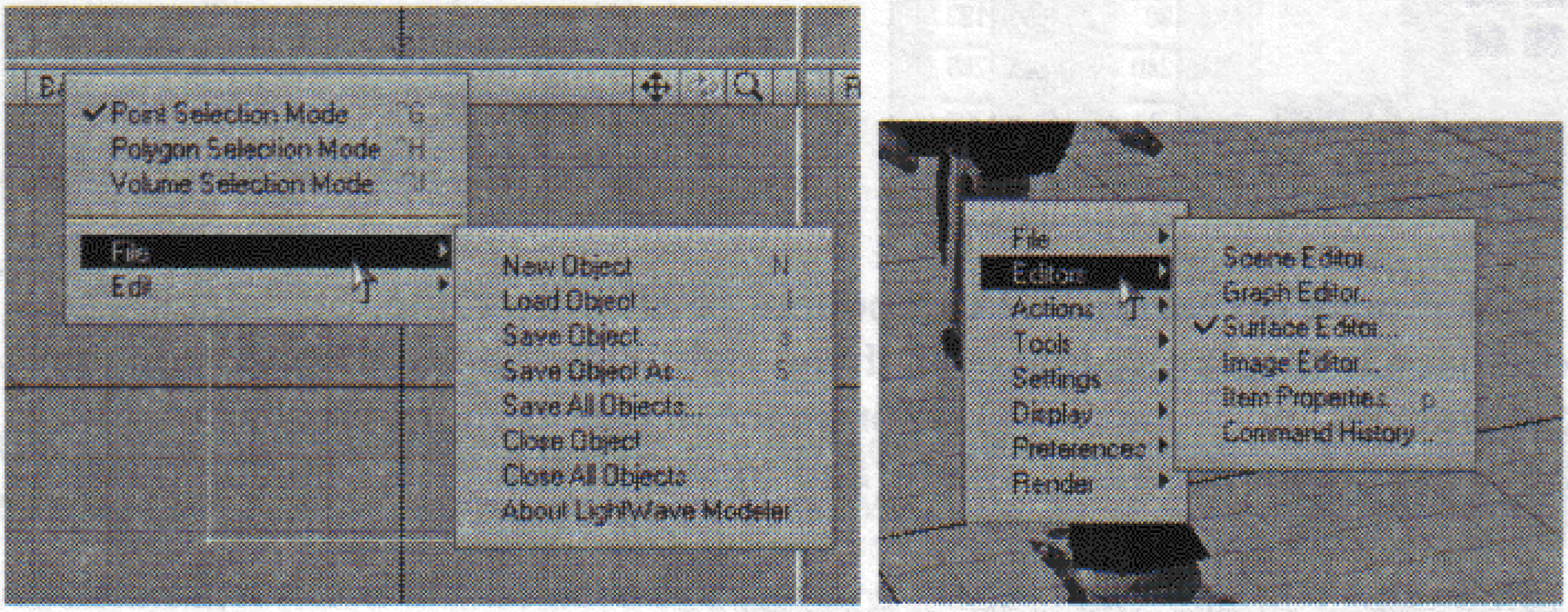
List windows that are hierarchical in nature can be reorganized by dragging items around in the list. Where appropriate, multiple selection is supported. See the following discussion on customizing the menus for examples on how to reorganize list windows.

NOTE

When editing menus and keyboard mappings, you can drag commands to the assignment window.

SPECIAL POP-UP MENUS

Both Layout and Modeler have special pop-up menus which are displayed by holding Ctrl + Shift and clicking your LMB, MMB or RMB with your mousepointer over a viewport—there are different menus for each mousebutton. These can be customized as discussed below.



Left: Modeler LMB menu. *Right*: Layout LMB menu

CUSTOMIZING MENUS

WARNING

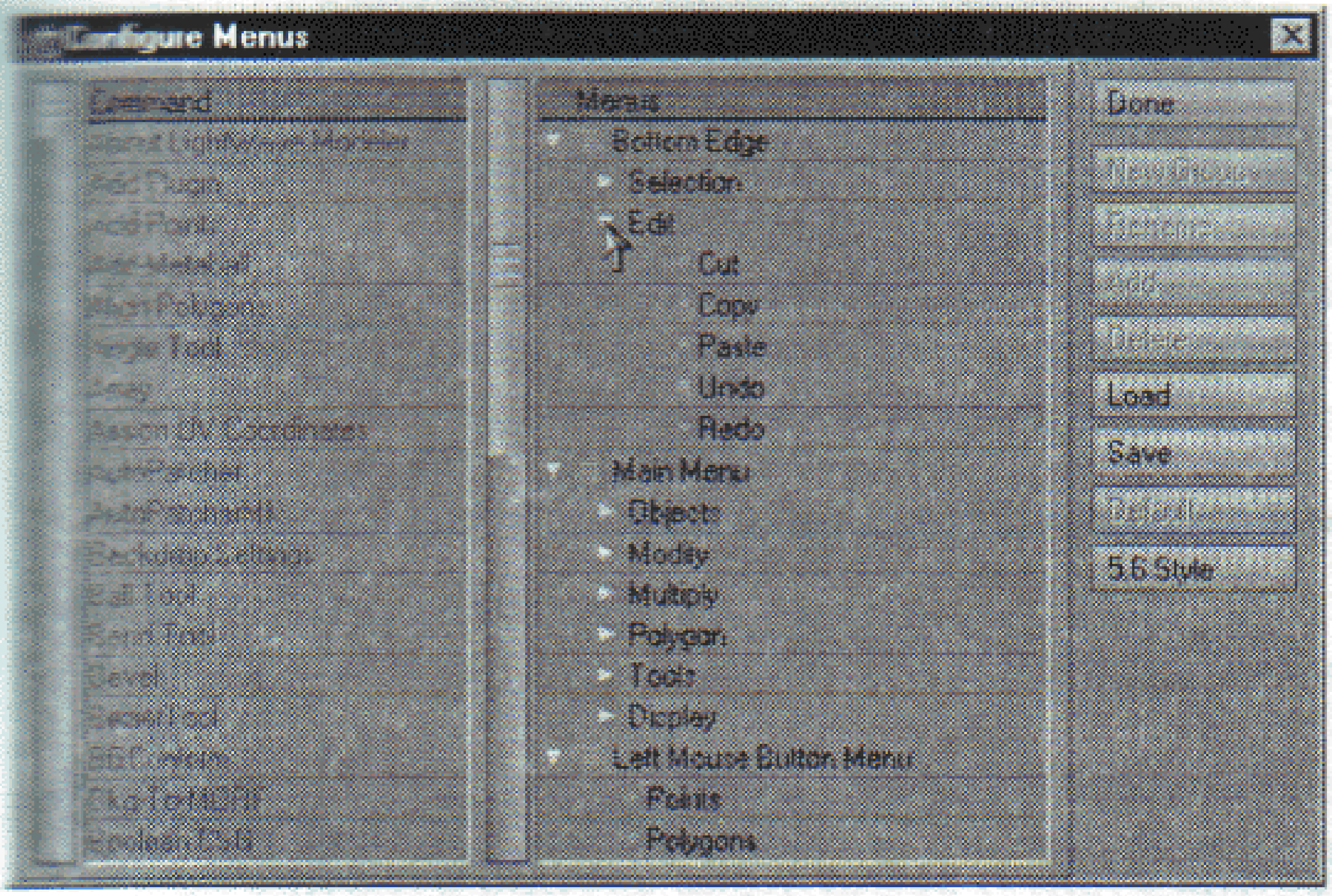
It is strongly suggested that you keep the default menu organization intact. Otherwise, technical support and manual utilization may become difficult. A better solution is to create a new menu tab and/or groups and place your frequently used tools in them.

LightWave menus are customizable. You can add, remove, group, and reorganize commands. The group can also have a definable name.

HINT

If you are having trouble finding a particular button, check the **Configure Menus** panel. Locate the function in the left window. If it is not ghosted, add it to a menu. If it is ghosted, you can either add it again or locate it in the right menu window. You can also take a similar approach in the **Configure Keys** panel.

Open the Configure Menus panel. (**Objects** > **Preferences** > **Edit Menu Layout** for Modeler or **Extras** > **Edit Menus** for Layout). There will be several main sections with sub-items beneath using indentation to show the hierarchical relationships. **Main Menu** items are the main tabs and related buttons for the main interface. **Bottom Edge** is for the Modeling module only and relates to the controls along the bottom edge of the screen. **Left**, **Middle**, and **Right Mouse Button Menu** (when available) will appear when the SHIFT + CTRL keys are held down along with the corresponding mouse button.



Modeler panel

Dots Cut indicate commands while arrows indicate groups of commands/sub-groups beneath it. If the arrow is pointing to the right ▸ Selection, the group's sub-menu items are collapsed and not visible. To reveal a closed group's sub-menu items, simply click on its right-facing arrow. To collapse an open group, click on its downward-facing arrow ▾ Main Menu.

First-Level Menu Items

First-level menu items must always be a group. Commands and sub-groups may exist on the second menu level. On the interface, a Main Menu group's name will appear as a heading above the grouped command buttons. If the group name is blank, no heading will appear. On the next menu level, commands or sub-groups can also be used. If a sub-group is used, it will appear as a pop-up menu on the interface.

Section Type (e.g., Main Menu)
    Group (Group name appears on row of horizontal tabs on the main interface)
        Command/Sub-group (Group name appears as a header above the group of commands)
            Command/Sub-group (Group name appears on pop-up button)

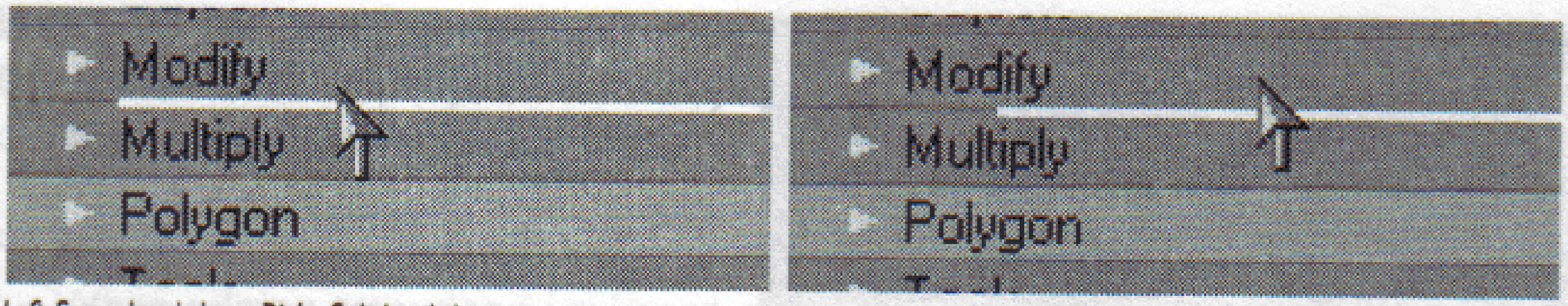
WARNING

Keep in mind open interface real estate. Do not add more menus than your maximum screen size can display.

Reorganizing Menus

You can reorganize entire groups or single commands by just dragging them in the Menus window. Place your pointer over the group/command and drag it up or down. You will see a thin line appear as you drag the group/command. Releasing the mouse button will drop the group/command at this point.

Now, if your pointer is at a position where the dropped item could be, say, a sub-item of a preceding group, the same level as the proceeding group, or maybe at the same level as the next group, you will see the line change its length as you drag. The position of the left end of the line determines the level of the drop.



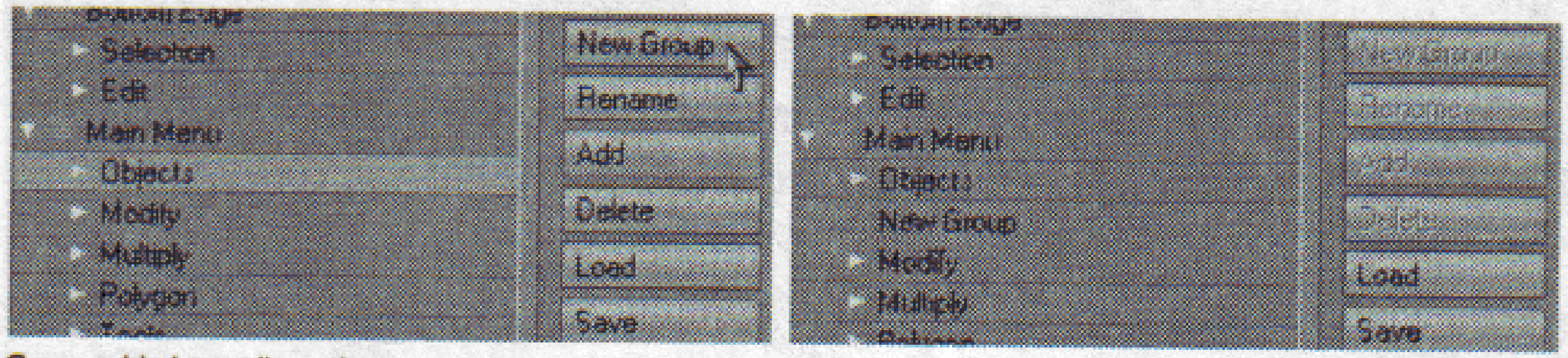
Left: Same level drop. *Right:* Sub-level drop

HINT

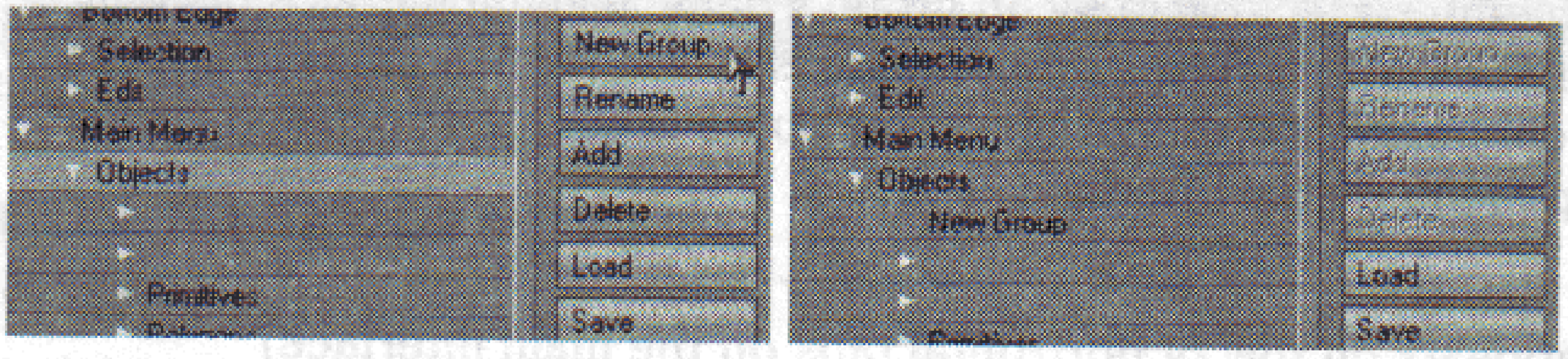
To place the group/command at a specific point in a group, make sure the group is uncollapsed before you begin dragging.

Adding New Groups

If a collapsed group is selected when you click the **New Group** button, it will be added at the same level as the group. If an uncollapsed group is selected when you click the **New Group** button, it will be added inside the group.



Group added to collapsed group



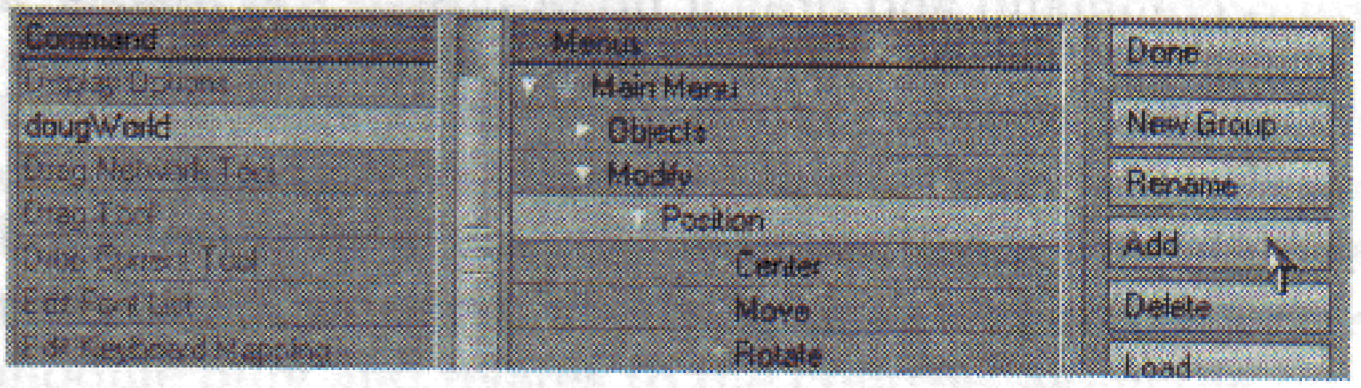
Group added to uncollapsed group

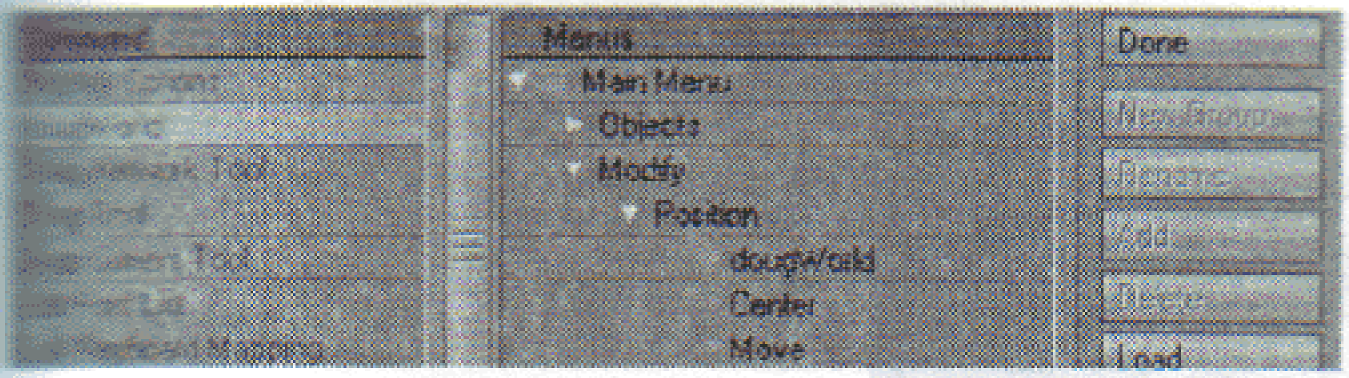
Renaming Menu Items

You can rename any menu item (including commands) by selecting it with your mouse and then clicking the **Rename** button.

Adding Commands

To add a command, select it in the (left) Command window and select the target position in the (right) Menus window. Then click the **Add** button. The command will be added under the selected target—inside the group if a group was selected.





NOTE

Commands that are already added to a menu appear ghosted in the Command window. That just indicates that the command already appears somewhere in your menus. However, the same command can still be added more than one time.

Deleting Menu Items

To delete a command/group, select it and click the **Delete** button.

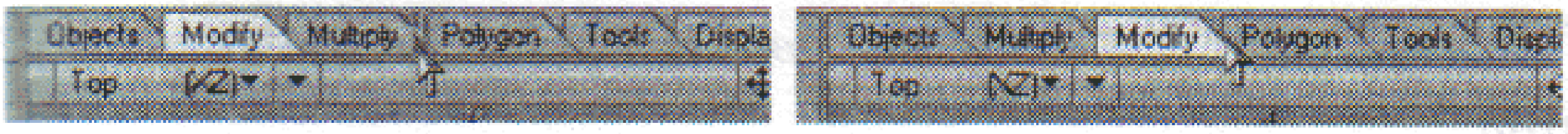
Maintaining Menu Sets

The **Load** and **Save** buttons allow you to retrieve and store *menu sets* that you develop. To restore the default menu configuration, click the **Default** button. If you'd like to have a menu set similar to LightWave 5.6, click the **5.6 Style** button.

NOTE

In Modeler, you can use **Objects > Preferences > Revert to Startup Preferences** to restore your initial settings; however, any changes since you first ran the application will not be reflected.

Arranging Menu Tabs



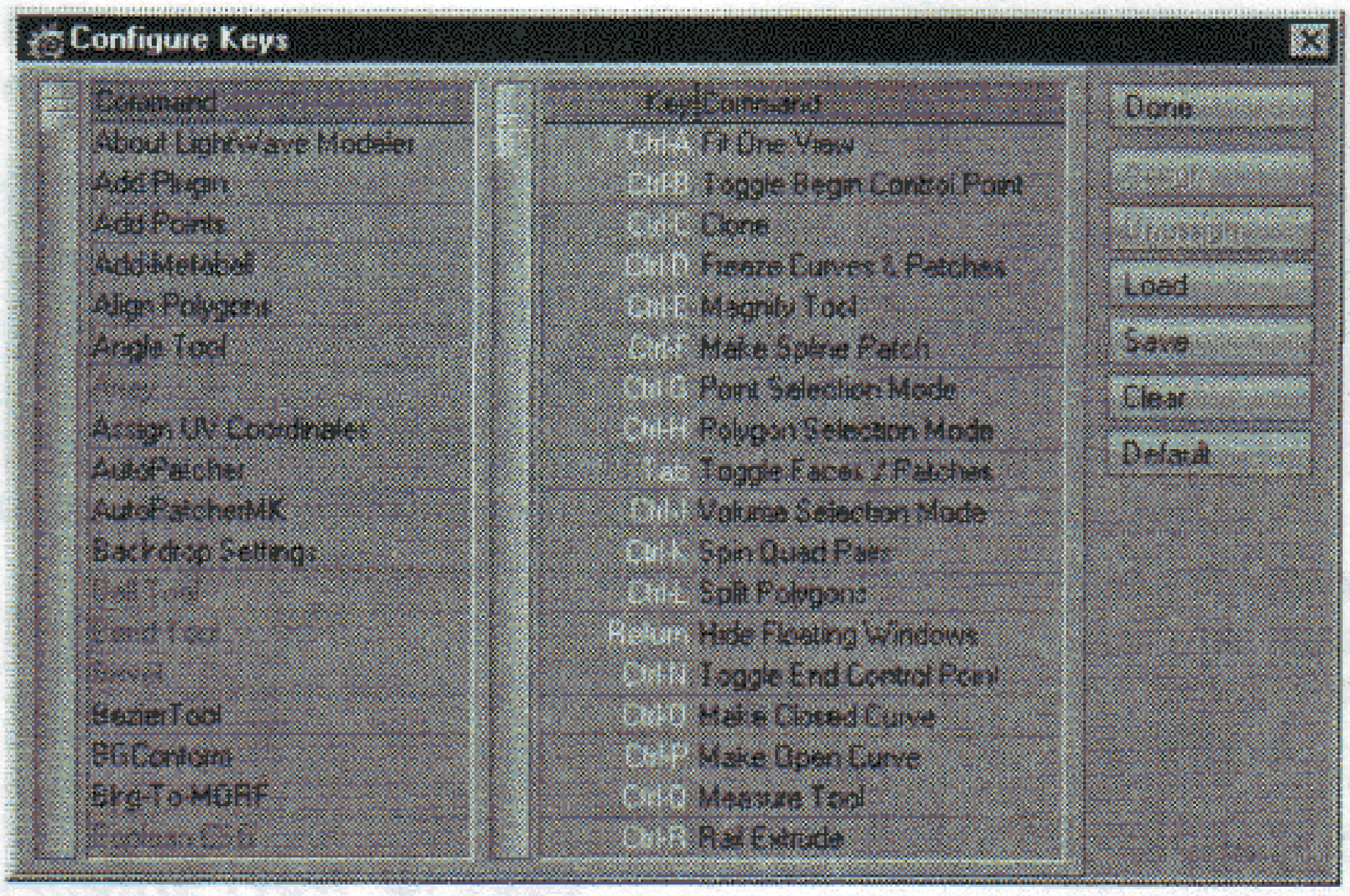
You can even reorganize the main menu tabs by simply dragging them left or right on the main interface. Before you release the mouse button you will see an insertion point market appear.

KEYBOARD SHORTCUTS

WARNING

It is strongly suggested that you keep the default keyboard mapping assignments and only make new assignment to unmapped keys. (Most of the function keys are open.) Otherwise, technical support and manual utilization may become difficult.

Like the menus, keyboard shortcuts can be configured to suit your own needs. To display the Configure Keys panel, choose **Objects** > **Preferences** > **Edit Keyboard Shortcuts** for Modeler or **Extras** > **Edit Keys** for Layout.



Command
About LightWave Modeler
Add Plugin
Add Points
Add Metaball
Align Polygons
Angle Tool
Array
Assign UV Coordinates
AutoPatcher
AutoPatcherMK
Backdrop Settings
Ball Tool
Band Tool
Bevel
BezierTool
BGConform
Blrg-To-MORF
Boolean CSG

Key	Command
Ctrl-A	Fit One View
Ctrl-B	Toggle Begin Control Point
Ctrl-C	Clone
Ctrl-D	Freeze Curves & Patches
Ctrl-E	Magnify Tool
Ctrl-F	Make Spline Patch
Ctrl-G	Point Selection Mode
Ctrl-H	Polygon Selection Mode
Tab	Toggle Faces / Patches
Ctrl-J	Volume Selection Mode
Ctrl-K	Spin Quad Pair
Ctrl-L	Split Polygons
Return	Hide Floating Windows
Ctrl-N	Toggle End Control Point
Ctrl-O	Make Closed Curve
Ctrl-P	Make Open Curve
Ctrl-Q	Measure Tool
Ctrl-R	Rail Extrude

Modeler panel

The window on the list contains a complete list of all assignable commands. The right window is a complete list of all keystrokes and assigned commands, if applicable.

To assign a command to key:

1 Select a command in the left window.

2 Select the target key in the right window.

3 Click the **Assign** button. This will overwrite any existing assignment.

To unassign a command to key:

1 Select the target key in the right window.

2 Click the **Unassign** button.

☞ ***NOTE***

Shortcut keys will appear on the right side of command buttons and are case-sensitive. So, "S" would be SHIFT + S.

Maintaining Key Mapping Sets

The **Load** and **Save** buttons allow you to retrieve and store *key mapping sets* that you develop. To restore the default assignments, click the **Default** button. The **Clear** button clears out all assignments, so use with caution.

☞ ***NOTE***

In Modeler, you can use **Objects > Preferences > Revert to Startup Preferences** to restore your initial settings; however, any changes since you first ran the application will not be reflected.

YOUR DISPLAY

Depending upon your computer platform and display cards, you have the option of running LightWave in a number of different screen resolutions and color depths. However, the larger your resolution and the more colors you are using, the slower your LightWave interface will be. LightWave does require at least a 16-bit display to function.

NOTE

Even though you may be using a lower number of display colors, you can still save 24-bit images when rendering your final output.

LSCRIPT

LScript is a high-level wrapper for the LightWave plug-in API. It encapsulates the complex underpinnings of the API away from the plug- in developer, allowing them to concentrate more fully on the task to be accomplished. LScript also provides added features not available in the plug-in API, making plug-in development faster.

Because LScript has its roots in the C language, the transition between scripting and native-language (binary) plug-in development is eased a great deal. Scripts written in LScript can often be ported into C with far less effort. This makes it possible to use LScript as a rapid prototyping tool for plug-in development.

Nearly all of the LightWave plug-in architectures have scripting capabilities through LScript.

LScript also provides a run-time system, allowing scripts to be compiled into an encrypted binary form that prevents modification or reverse- engineering. Facilities for timed or counted execution are also provided by the run-time system.

Most important, LScript is a *virtual machine* system. Scripts written on one platform should work directly and immediately on any other platform supported by LightWave. This differs from traditional plug-in development in that each platform must have its own compiler, and each plug-in must be compiled and maintained on that platform. Script are platform independent.

Documentation is included on the LightWave CD.

chapter 4
Tutorials

CHAPTER 4: TUTORIALS

This chapter contains several tutorials to help you learn about LightWave 3D. These were designed to help you touch and play with combinations of tools and features to see how they work together.

☞ **NOTE**

These tutorials only being to scratch the surface of what you can do with LightWave. Please check our web site (www.newtek.com) regularly for more tutorials.

TEST RENDERS

You will often be asked to do *test render*. This simply means render the current frame to see the current state of your scene. You can choose **Actions > Render > Render Current Frame** or simply press the F9 key. Your choice.

You should make sure that **Image Viewer** is selected as your **Render Display** on the Render Options panel (**Actions > Render > Render Options**). If you want to watch LightWave *draw* the image make sure **Show Rendering in Progress** also on the Render Options panel is active. Unless directed to render a sequence of frames, **Auto Frame Advance** should not be active.

Also, after completing the render, if the Render Status window is open, you will need to close it before proceeding.

FRONT PROJECTION IMAGE MAP

We'll be using a Front Projection Map on an object shaped like the tub's ledge and have a three-dimensional spraybottle object sitting on top of the ledge.

☞ **NOTE**

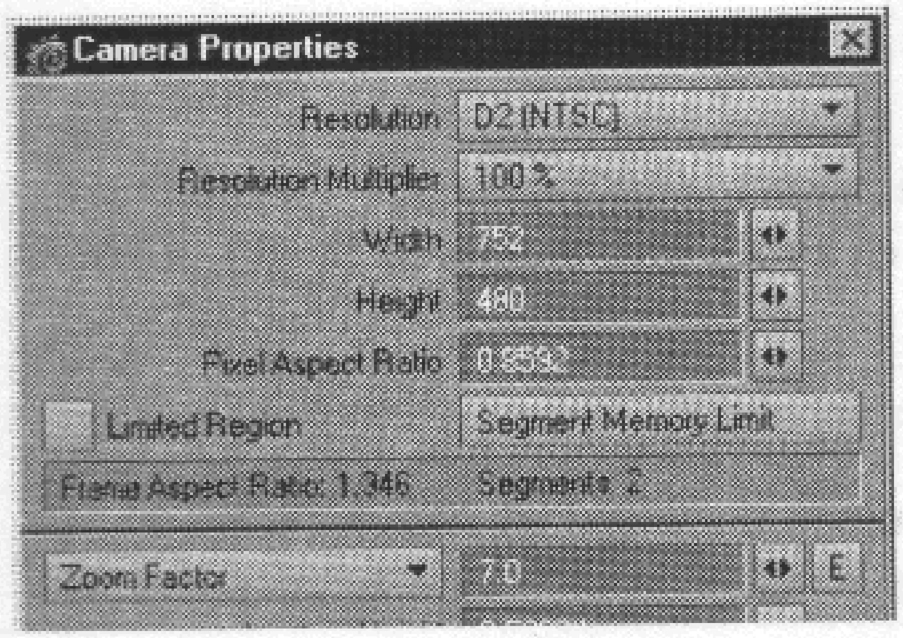
This is a very long tutorial, so you may want to save your object periodically along the way.

1 Clear the scene and choose **Settings > Compositing**.

2 From the **Background Image** pop-up menu, load the SHOWERTUB image from the TUTORIAL subdirectory. The image is a nice shot of a bathtub.

3 Select camera and open its properties panel.

4 Set the **Zoom Factor** to 7, which will better match the camera lens used to take the image. Also set the **Resolution** to **D2 (NTSC)** because the image was taken from a video tape which uses vertically-stretched pixels (i.e., NTSC video). On your computer monitor the image will seem somewhat squashed.



Camera Properties

Resolution: D2 (NTSC)

Resolution Multiplier: 100 %

Width: 752

Height: 480

Pixel Aspect Ratio: 0.8592

□ Limited Region

Segment Memory Limit

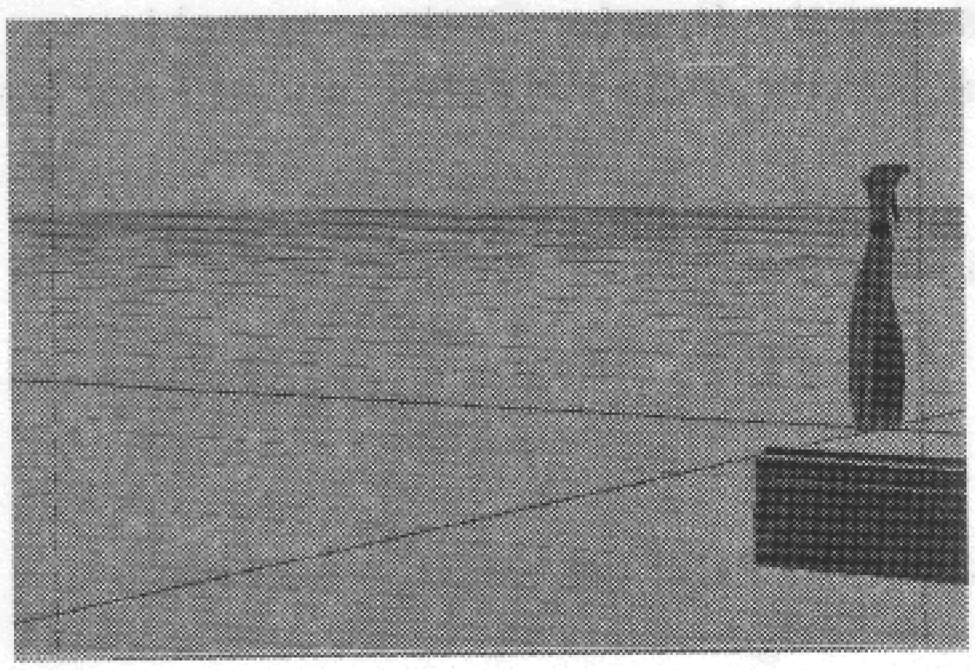
Frame Aspect Ratio: 1.346

Segments: 2

Zoom Factor: 7.0

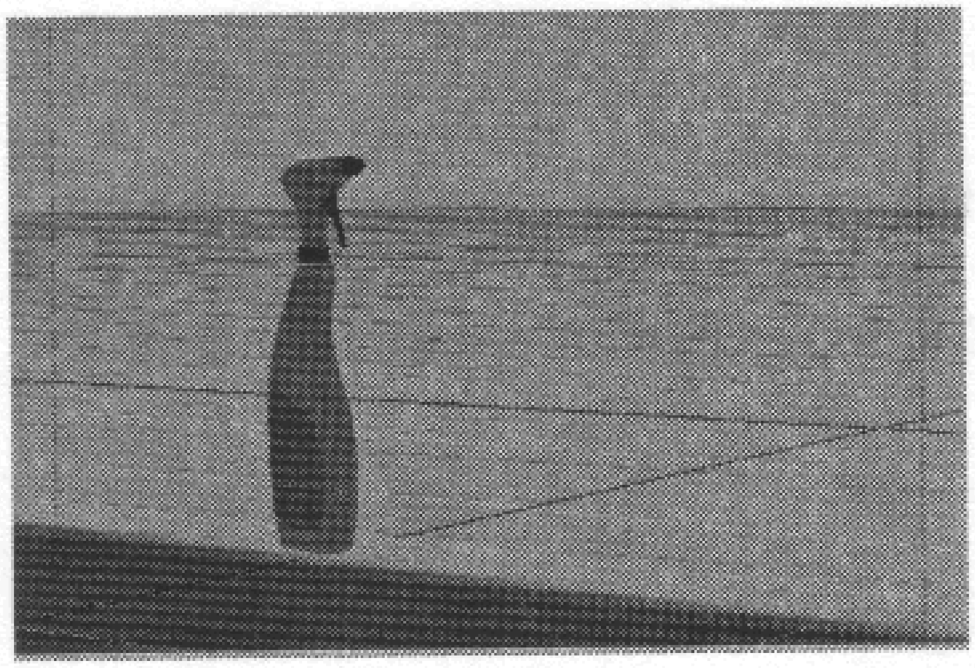
5 Switch to the Camera View. Then, move the camera to XYZ 10.553m, 1.277m, 4.871m and rotate it to HPB 235.3, 3.7, 0. Create a keyframe.

6 Load the SPRAYBOTTLE.LWO and TUBREND2.LWO objects from the TUTORIAL subdirectory. The objects will load into their default position and be practically out of camera range. One of the hardest parts of doing a scene like this is moving the 3D objects so that they appear in the right location and orientation.



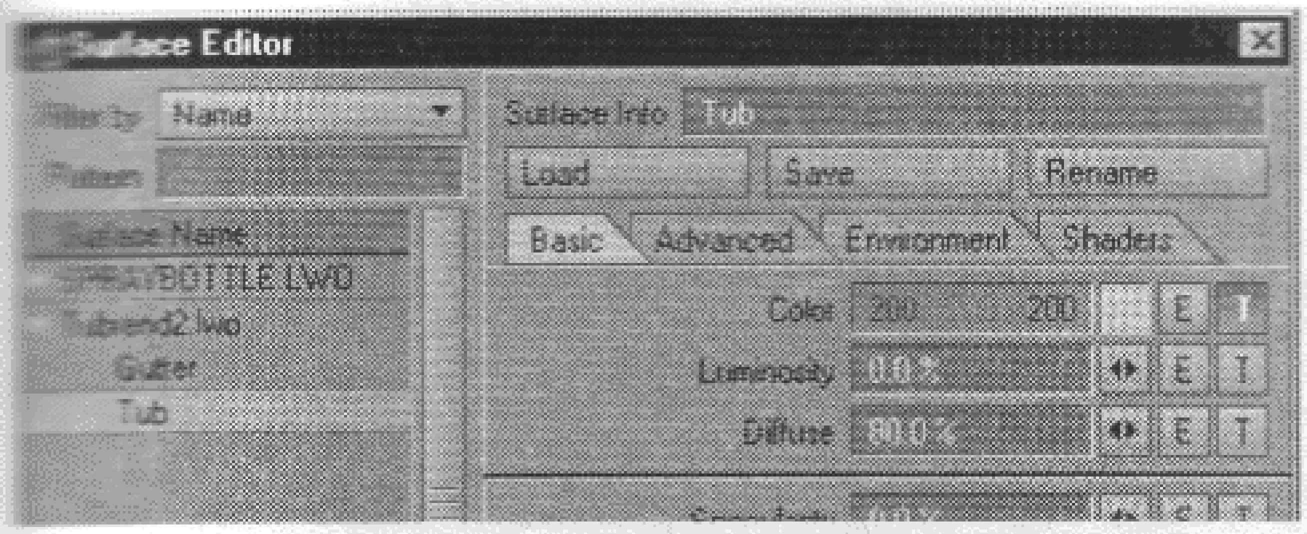
7 Select the Spraybottle and move it to XYZ 4.549m, .02m, 0m. Then, rotate its heading to 111. Make a keyframe.

8 Select the TubRend2 object and move it to XYZ 10.983m, -.19m, 5.855m. Rotate its heading to 182.8. Size it to XYZ 5.244, 5.244, 10.5. Make a keyframe. The TubRend object was a little too small so we sized it up a bit and stretched it out.



9 Open the Scene Editor panel and make the display mode for the TubRend2 object **Wireframe**.

10 Open the Surfaces Editor (**Settings** > **Surface Editor**) and select the Tub surface.



Filter by: Name

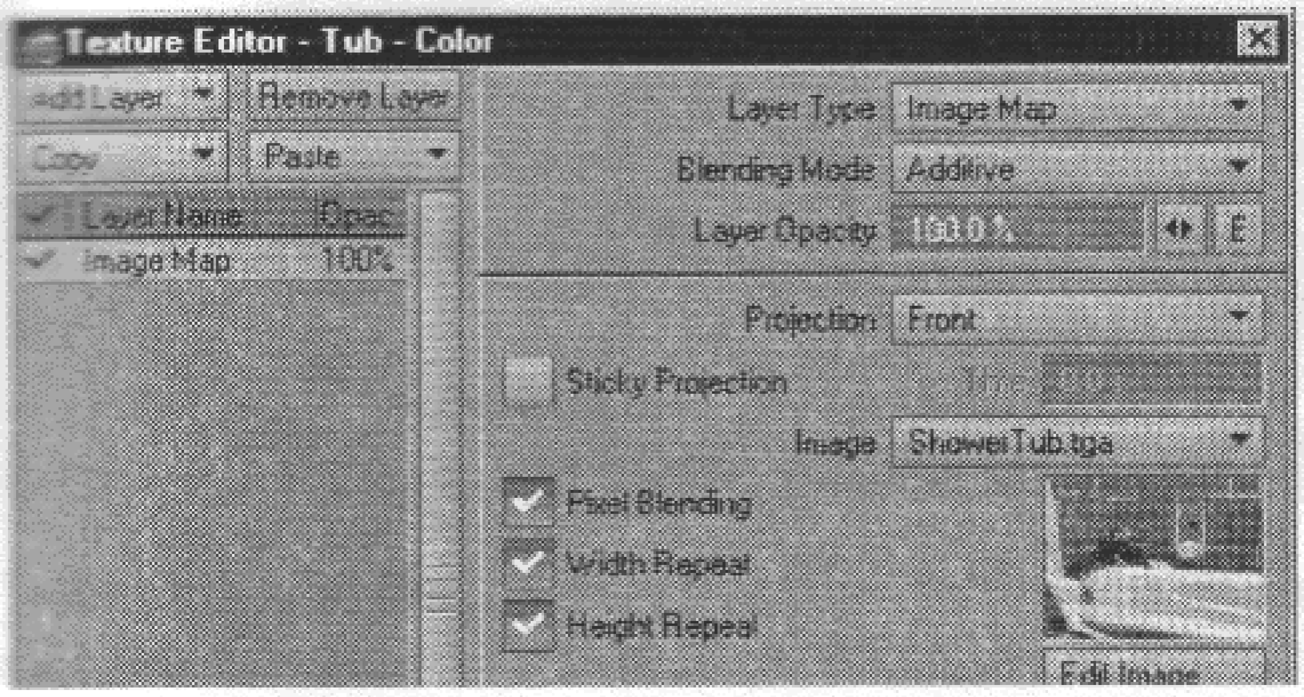
Surface Info: Tub

Color: 200 200

Luminosity: 0.0 %

Diffuse: 80.0 %

11 Click the **Color** Texture button. Note that the ShowerTub image is being used on a Front Project map. In the chapter on surfacing, you learned that this means the surface picks up the background image.



Layer Type: Image Map

Blending Mode: Additive

Layer Opacity: 100.0 %

Projection: Front

□ Sticky Projection

Image: ShowerTub.tga

☑ Pixel Blending

☑ Width Repeat

☑ Height Repeat

12 Click Use Texture.

13 Set **Luminosity** to 50%. Although we could add a set of lights to mimic the lighting used in photographing the actual bathtub, for purposes of this exercise we will cheat a little and artificially brighten the object.

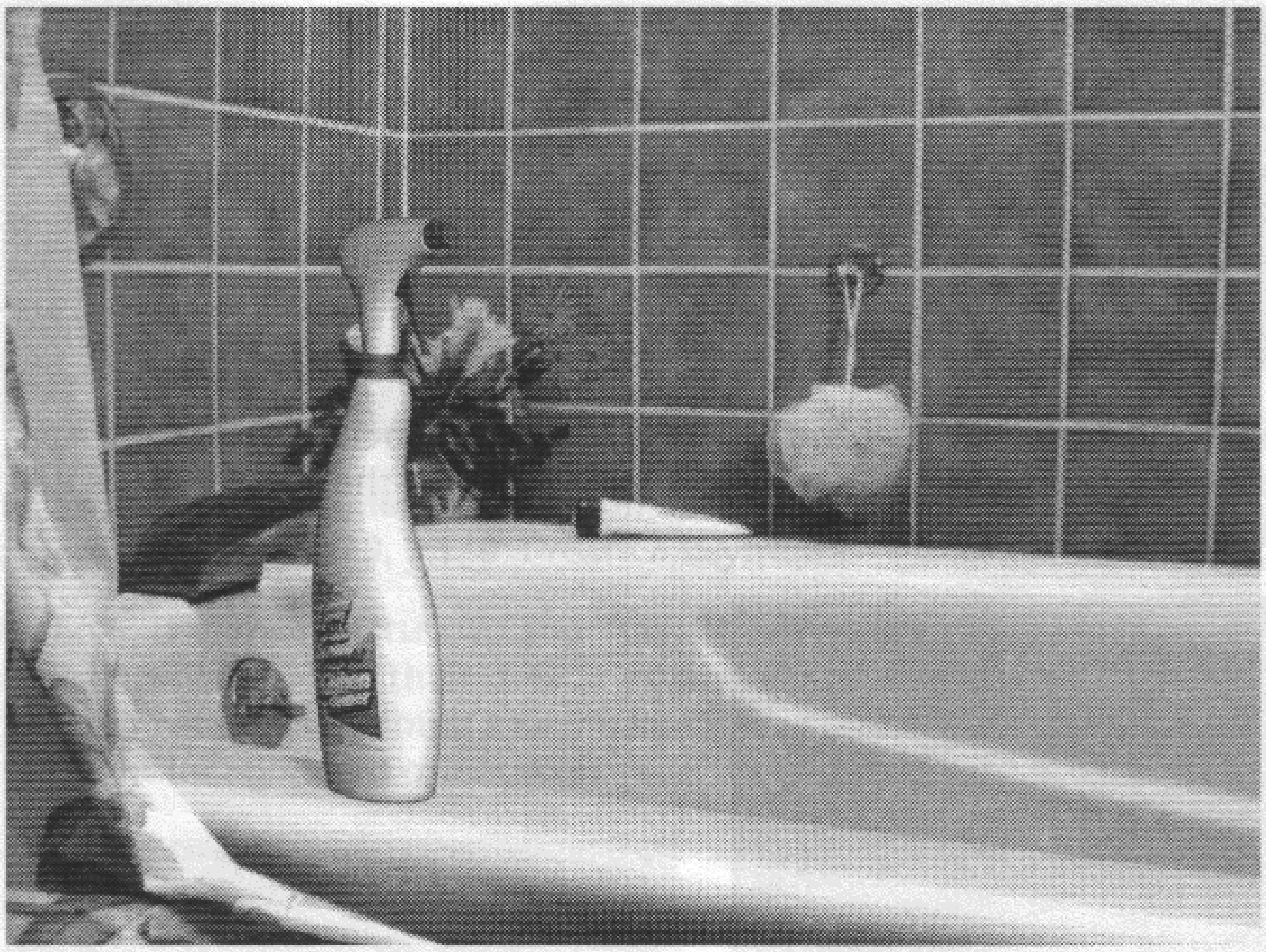
14 Select the light and move it to XYZ 11.074m, 3.310m, 7.313m. Then, rotate it to HPB -137.5, 16.2, .4. Make a keyframe. This will approximate the lighting used in the image.

15 Choose **Settings** > **Light Type** > **Spot**. Open the Light Properties panel and change the **Shadow Type** to **Shadow Map**. Click the **Global Illumination** button on the panel and make sure **Enable Shadow Maps** is active.

☞ **NOTE**

We could have also changed the **Light Type** on the Light Properties panel and you can also access the Global Illumination panel by choosing **Settings** > **Global Illum**.

16 Do a test render.



Wow! The spraybottle is casting a shadow on the tub's ledge and looks like it is sitting on the ledge. But wait! There's more.

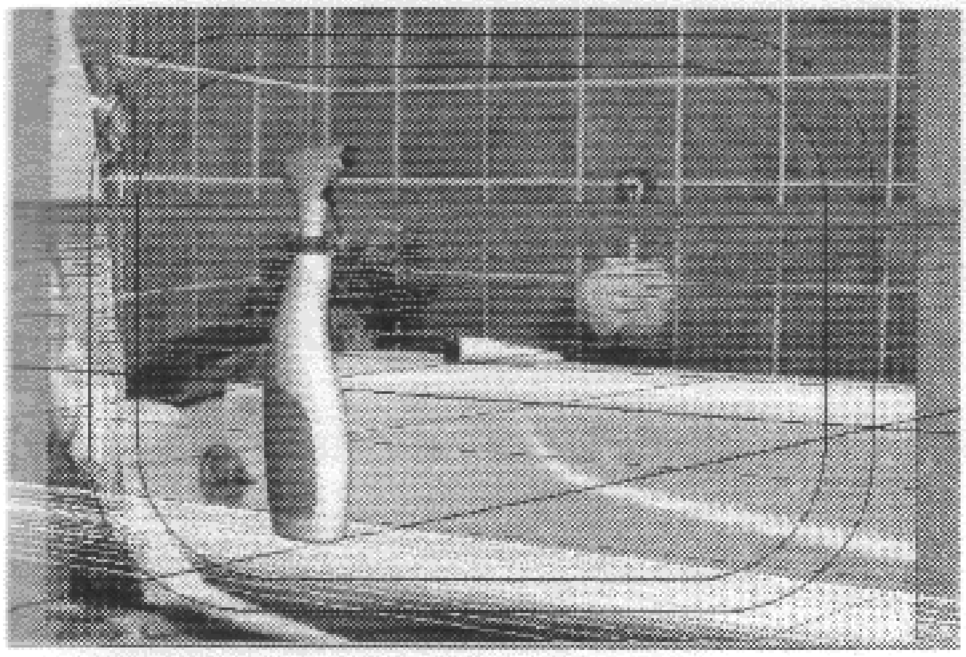
17 Open the Surfaces Editor and change the Tub surface's **Reflection** to 30%. Go to the Environment tab and make sure that the **Reflection Options** is set to **Ray Tracing + Backdrop**.

18 Now, go to the Render Options panel (**Actions** > **Render** > **Render Options**) and activate **Ray Trace Reflection** on the Rendering tab.

19 Do another quick render (F9). You should now see a slight reflection of the spraybottle on the tub's ledge!

You might notice that you can see the tub object through the curtain on the left. We could edit the tub object so that it ended at the curtain or even add a curtain object with front projection. However, this part of the image would likely be in the video overscan area and we may not need to do anything. We can check this.

20 Open the Display Options panel (**Extras** > **Display Options**) and activate **Show Safe Areas**. Also set the **Camera View Background** to **Background Image**. Activate **OpenGL Textures**, if it isn't already. Close the panel. You will see the curtain is nearly entirely in the overscan area and would likely not be seen or noticed.



NOTE

Try loading the SprayBottle.lws from the Tutorials subdirectory. This is a more complex version of the Scene you just created. In this Scene, a second spraybottle pops from inside the tub and moves around through the use of bones. Multiple lights were used to get a more realistic look. Also, a series of parented Null objects were used to move, rotate, and size the objects into position.

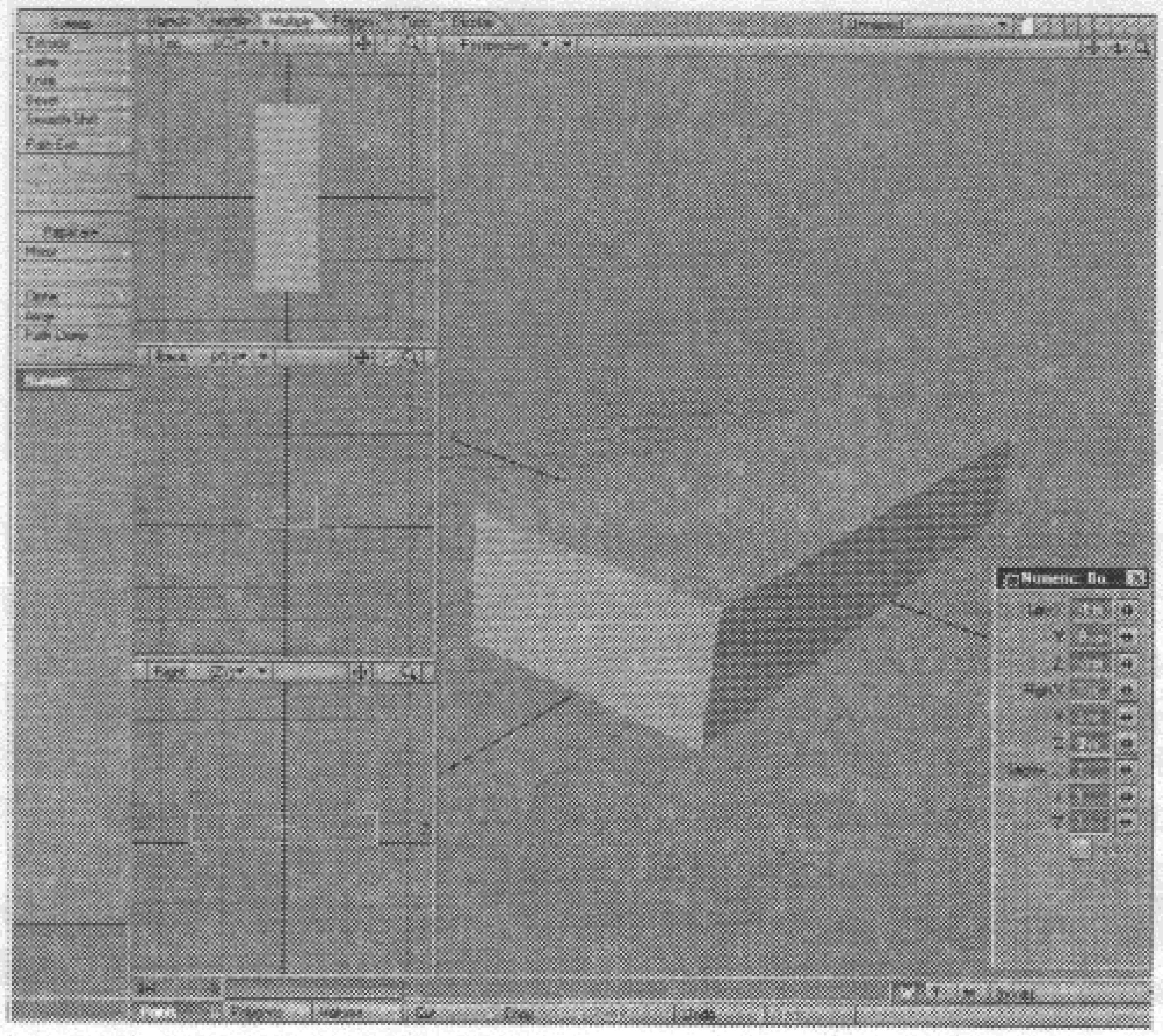
In this tutorial we will create and surface a simple car. Pay attention to the tools used since these are the most common tools you will use on other projects.

THE PECORA LOVE MACHINE

1 Open Modeler and in the Display Options panel (D) under the Units tab, set your **Unit System** to **Metric** and your **Default Unit** to **Centimeters**. On the Layout tab, change the **Layout** setting to **3 Left, 1 Right**. Set the viewports to match the screenshot below. Also, set the Right viewport to be independent on zoom and center.

2 Zoom out (,) until your grid is at 1m. (This value will be in lower left-hand corner of the modeler window.)

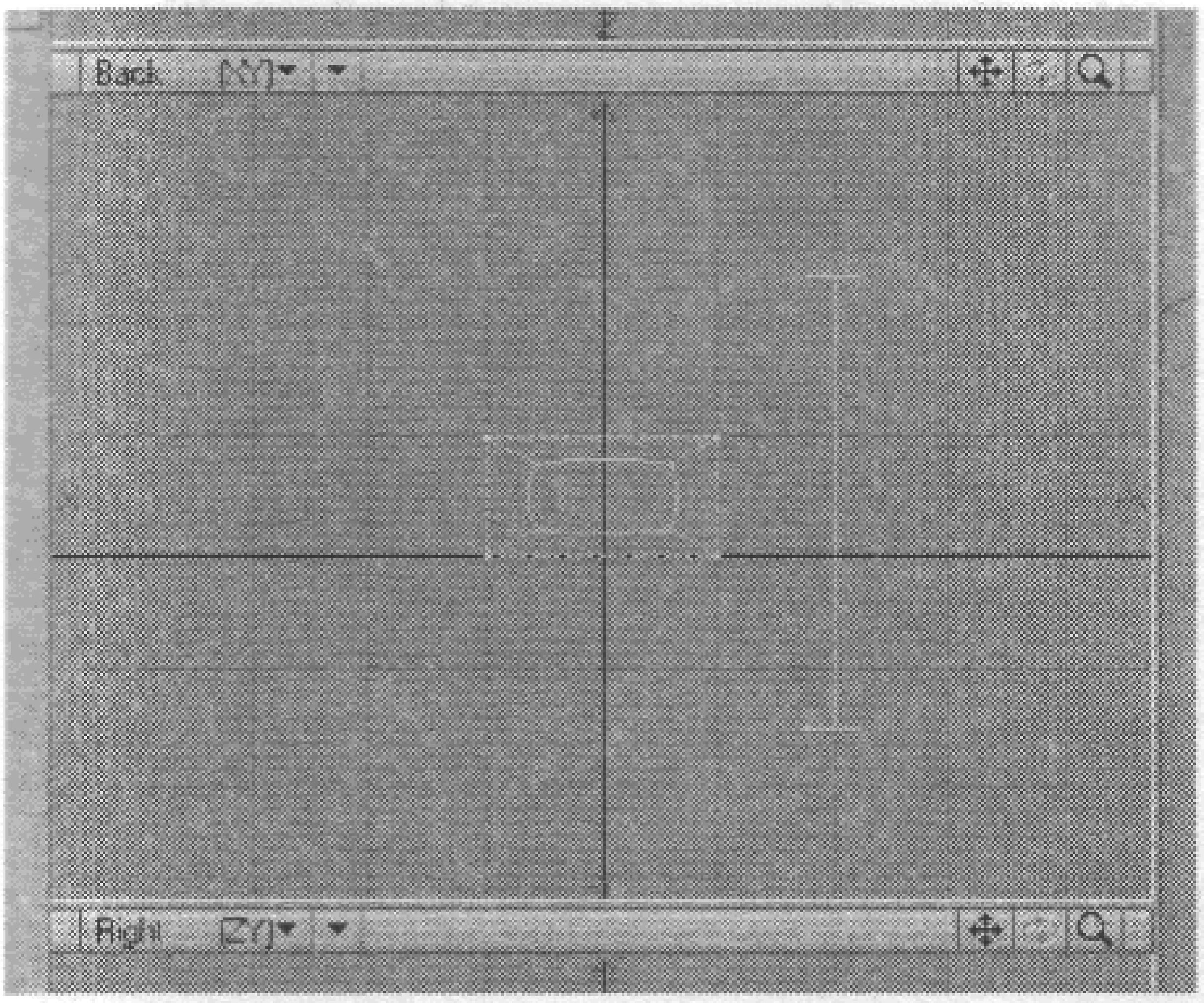
3 Select **Objects** > **Box** and then choose **Objects** > **Numeric** to open the Numeric panel. Select **Activate** from the **Actions** pop-up menu and enter the following dimensions: Low XYZ = -1m, 0m -3m; High XYZ = 1m, 1m, 3m. Deselect the **Box** tool to make the box.



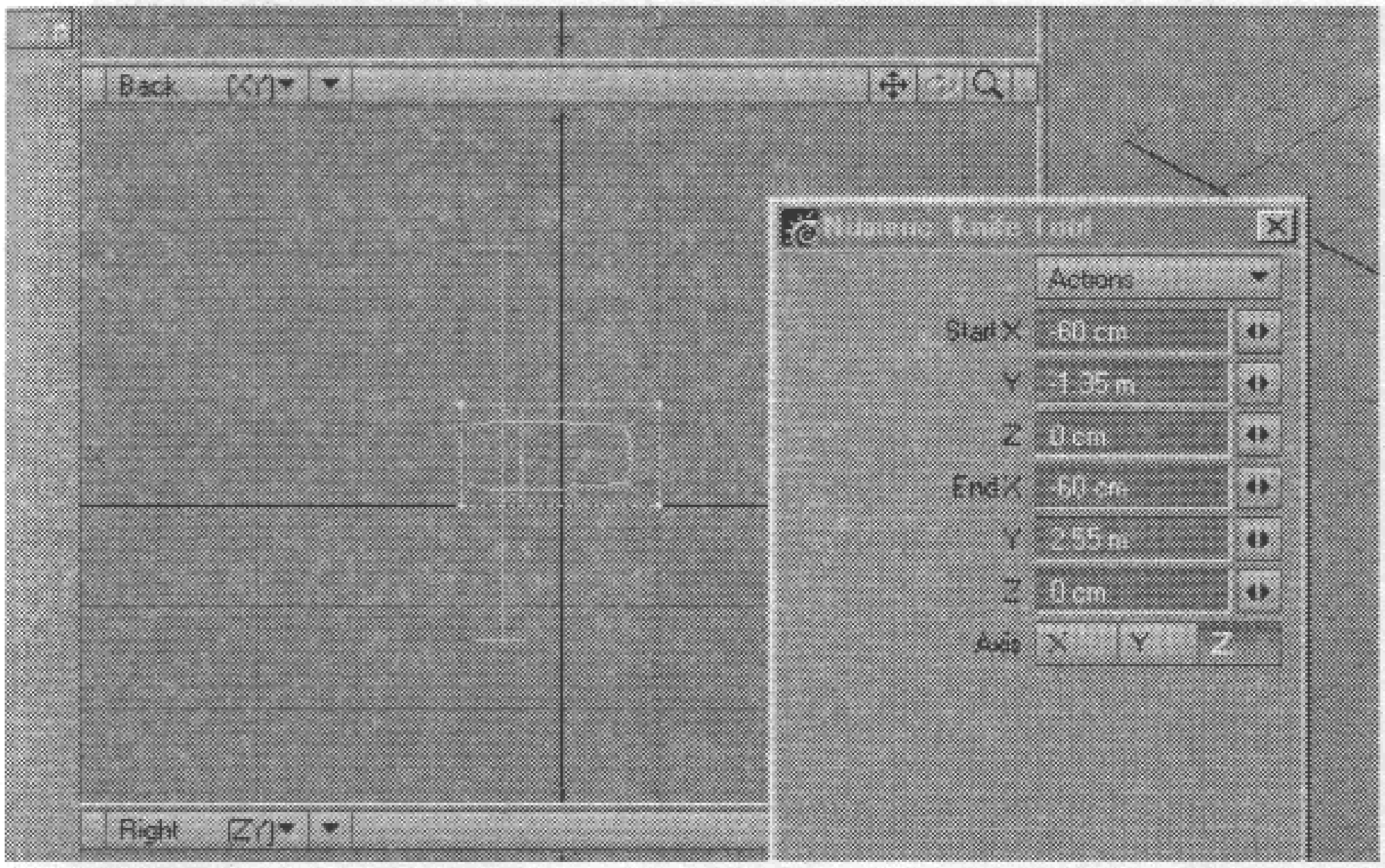
This is how your initial setup should look

4 Press the TAB key to activate the SubPatch mode.

5 Now let's make some slices in the box to give us some geometry to work with. Select **Multiply** > **Knife**. In the Back viewport, drag out a vertical line to the right of your box.



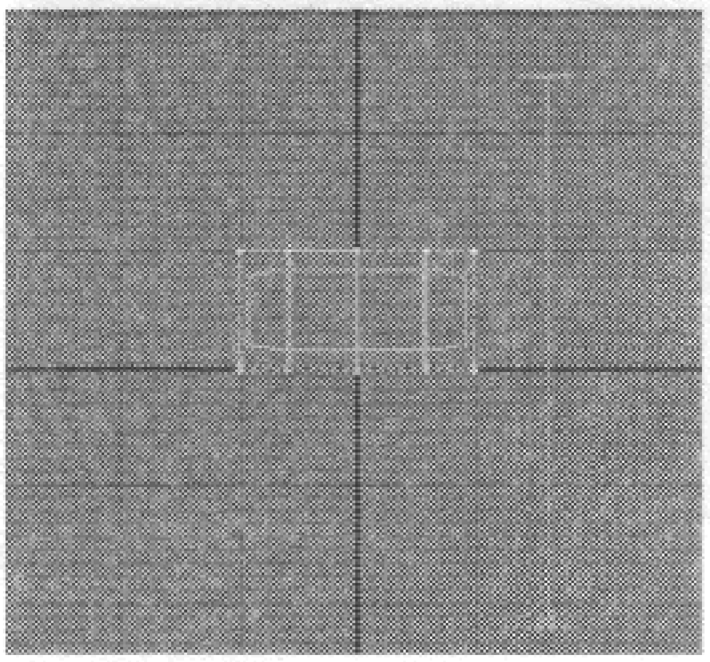
6 Drag the slicing line to -60cm on the X axis—use the X values in the Numeric panel for reference. As you drag, notice how the object previews the cut in real-time.



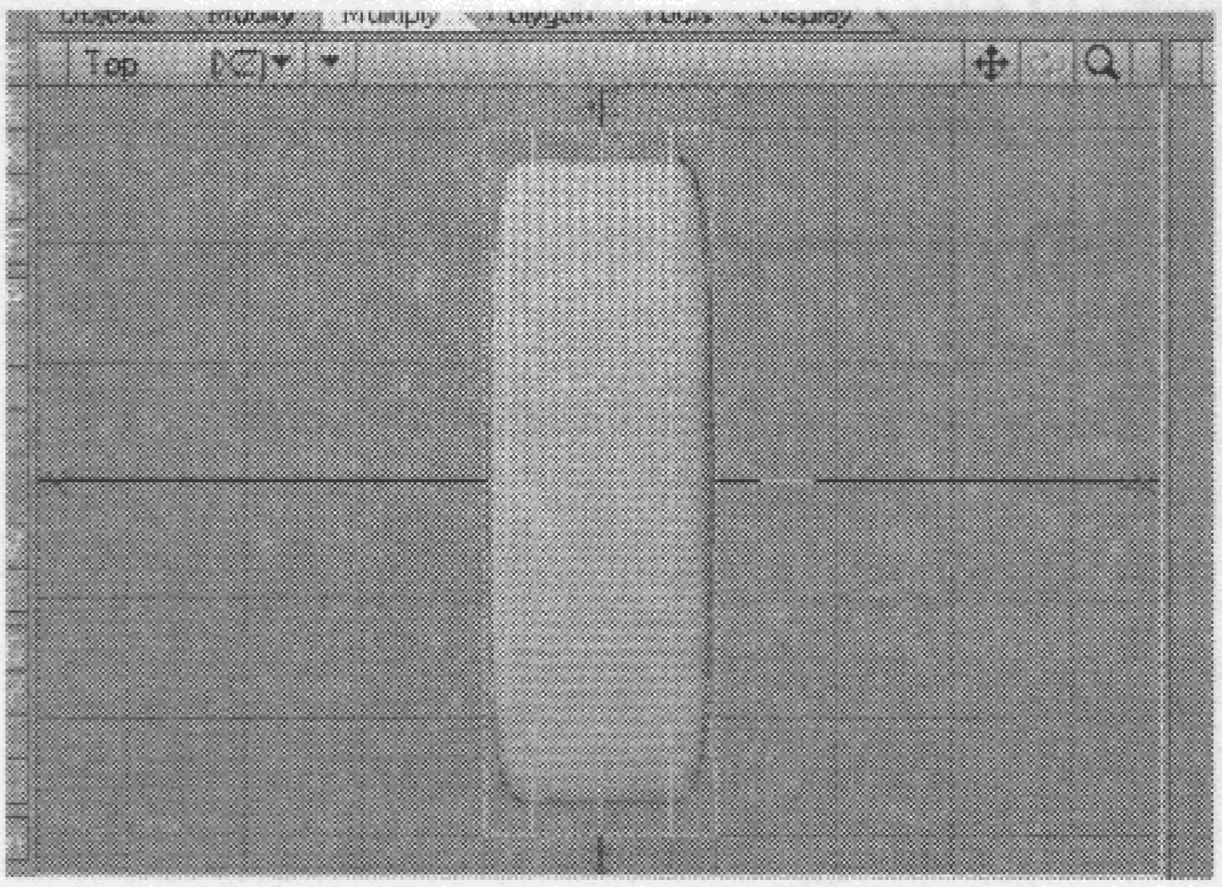
Start X -60 cm
Y -1.35 m
Z 0 cm
End X -60 cm
Y 2.55 m
Z 0 cm
Axis X Y Z

Click your RMB to make the slice.

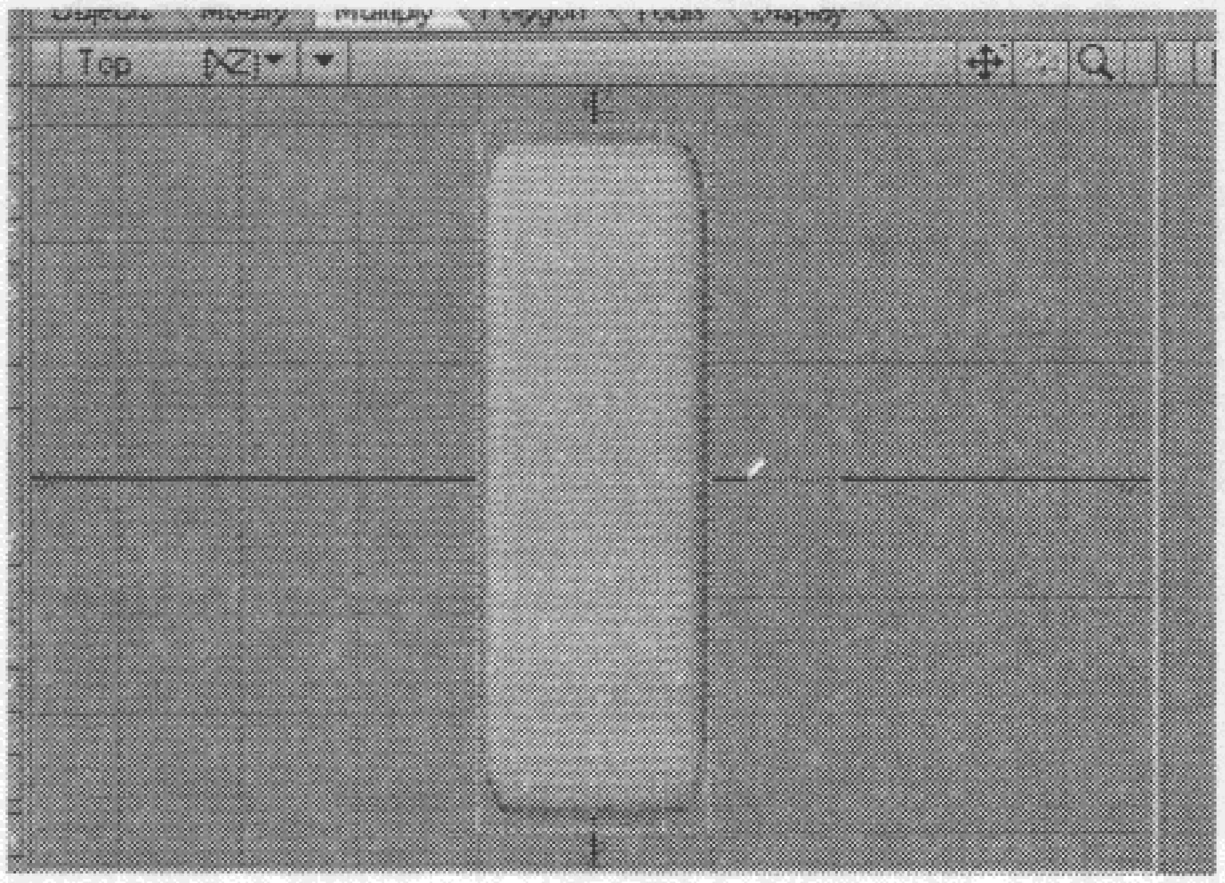
Next, make similar cuts at X = 0 and 60cm. There should now be three segments along the X axis.



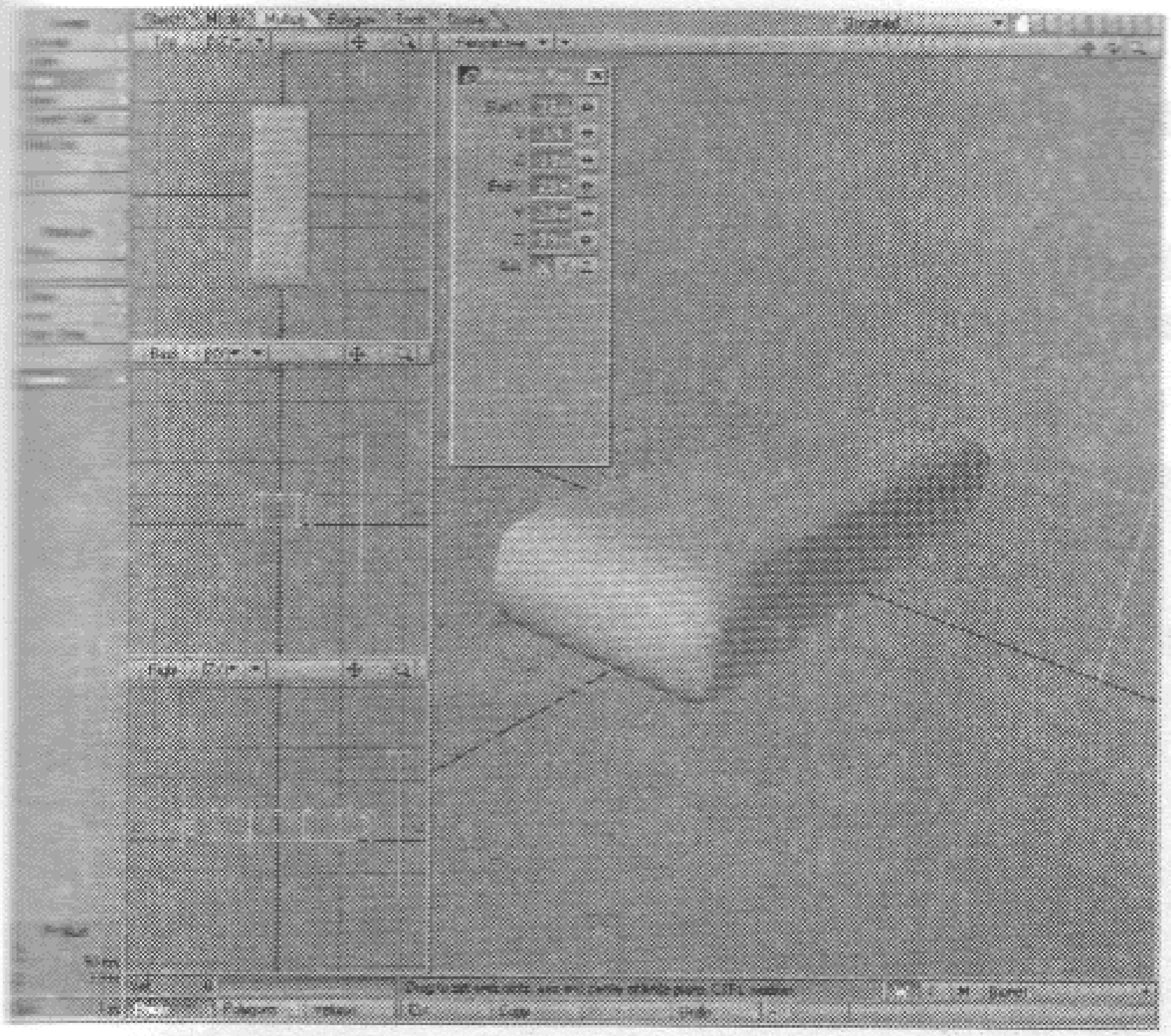
6b If you look in the Top viewport you will see the slicing line as a short blue line with a dotted blue line extending from it along the Z axis.



This dotted line represents the cutting directions. In the Top view, click on the tip of the blue dotted line and drag it around until it is aligned with the X axis. Keep in mind two things: first, the change happens in 90 degree increments; and second, the length of the dotted line may also change. This is fine. Length doesn't really matter here ;-)



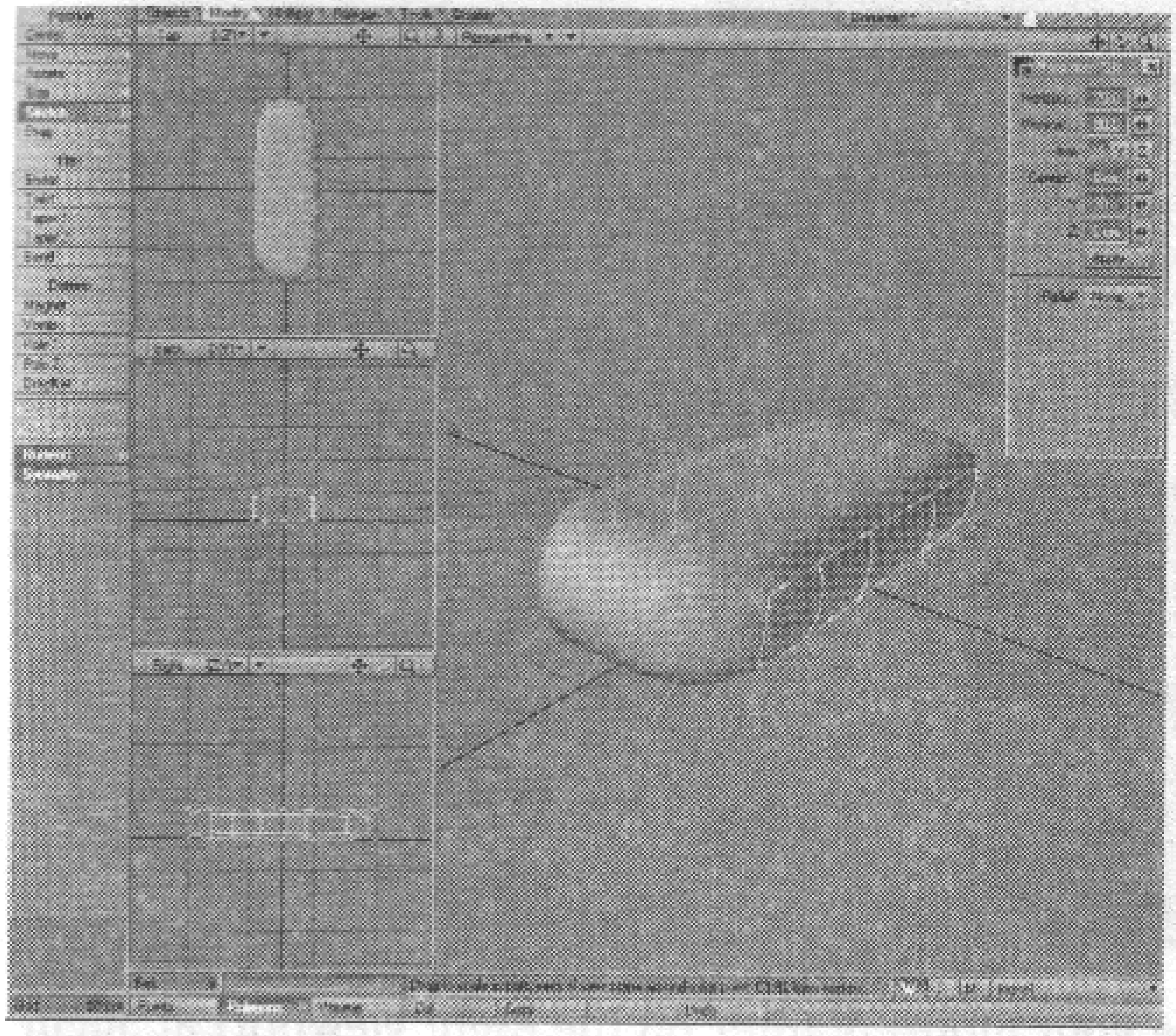
7 Using the Right viewport and the technique above, make slices along the Z axis at -2m, -1m, 1m and 2m. Deselect the **Knife** tool when you are done.



Using the Knife Tool

8 Now it is time to start shaping the box. Choose **Modify** > **Symmetry**. With **Symmetry** active, anything we move, select or affect on one side of the X axis will be moved, selected or affect on the opposite side.

9 First, let's select the polygons along the side of our box. Switch to **Polygon** Selection mode (CTRL+H), if you are not already in it. In the Perspective viewport, drag across the side of the object selecting the five polygons aligned with the Z axis. Since **Symmetry** is active, the five polygons on the opposite side are also selected. You should also note that since the Perspective viewport is using a shaded Rendering Style (Display Options), we don't need to worry about selecting polygons that are behind.

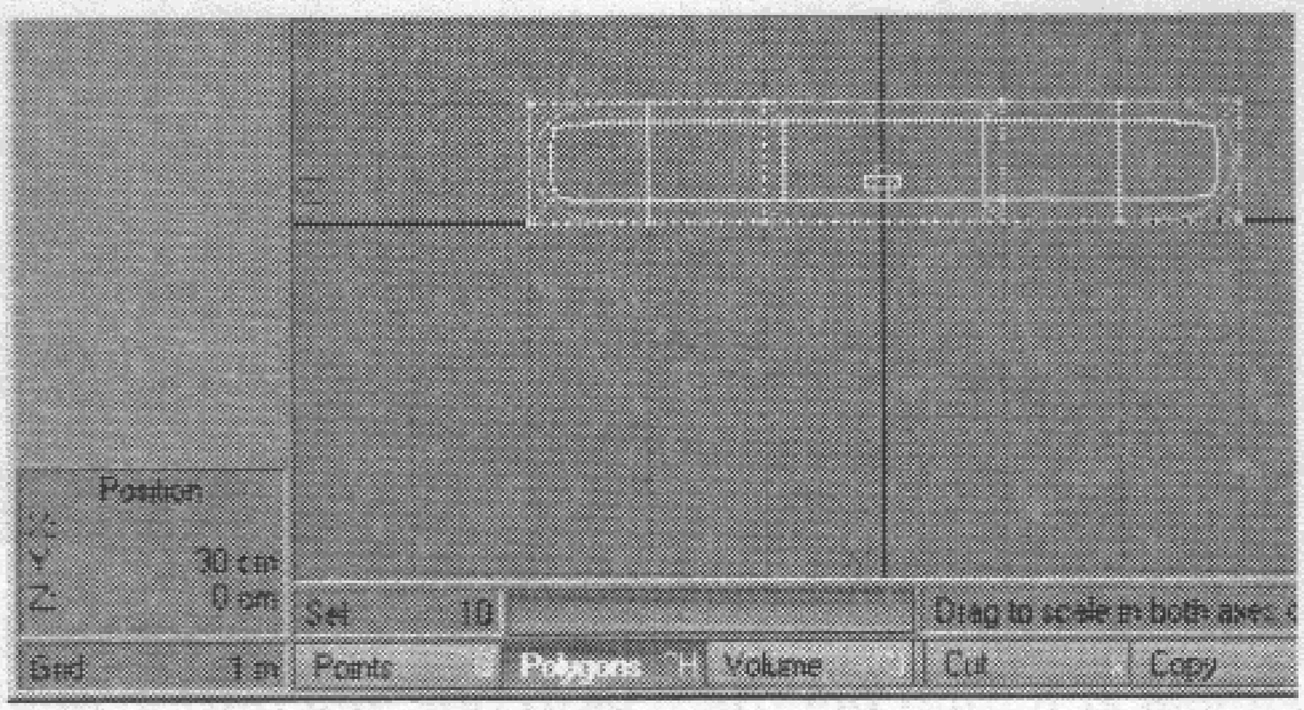


Selecting the side Polygons

☞ **NOTE**

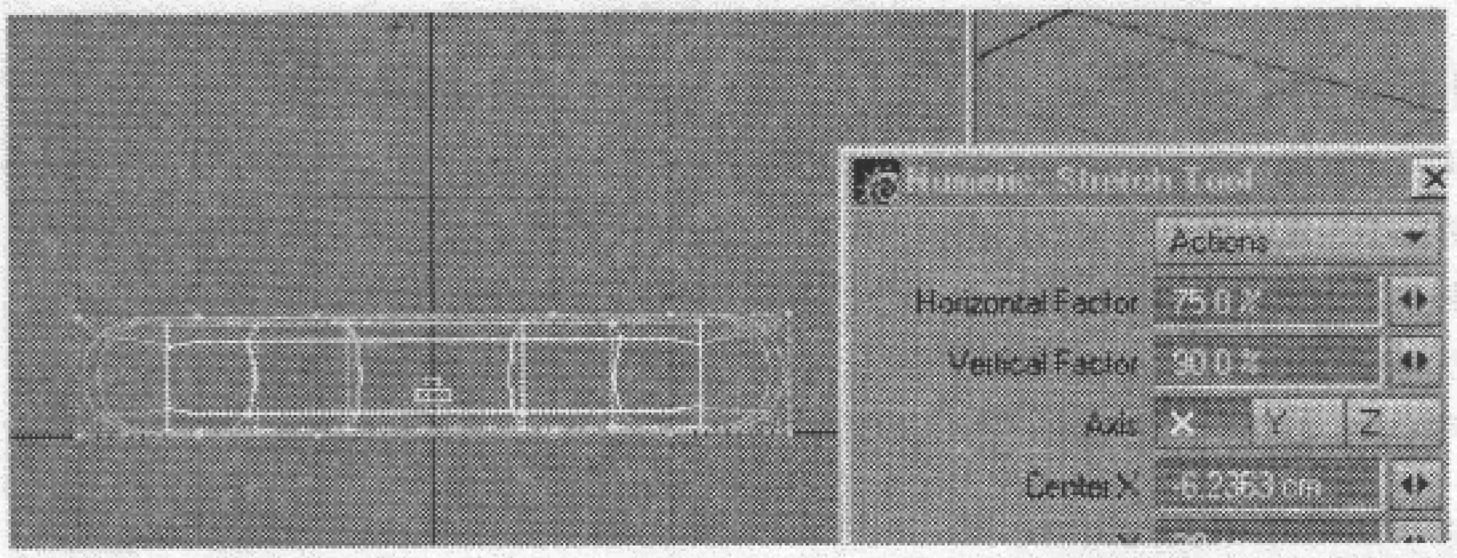
Holding down the ALT key while dragging will allow you to rotate the preview window for a better view.

9b Select **Modify** > **Stretch** to activate the **Stretch** tool. In the Right viewport, place your mousepointer at Y = 30cm. The information display in the bottom left corner can be used for reference.



Position
X
Y 30 cm
Z 0 cm
Sel 10
Drag to scale in both axes

Using the Numeric panel for reference, stretch these polygons 75% Horizontal and 90% Vertical by dragging your mouse left and down.

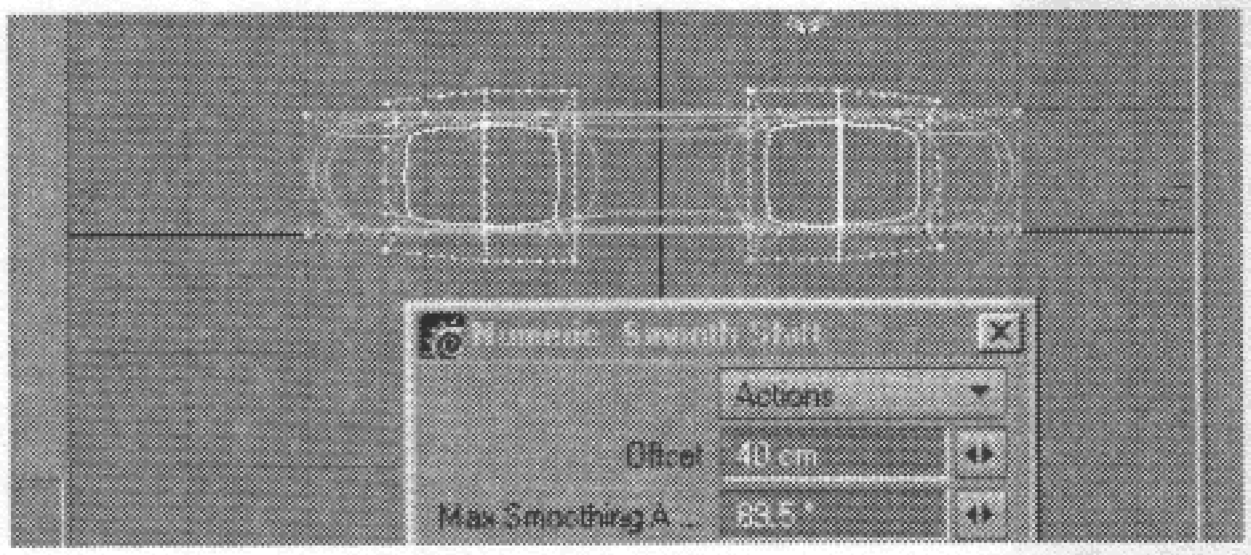


Numeric: Stretch Tool
Horizontal Factor 75.0 %
Vertical Factor 90.0 %
Axis X Y Z
Center X -6.2353 cm

☞ **NOTE**

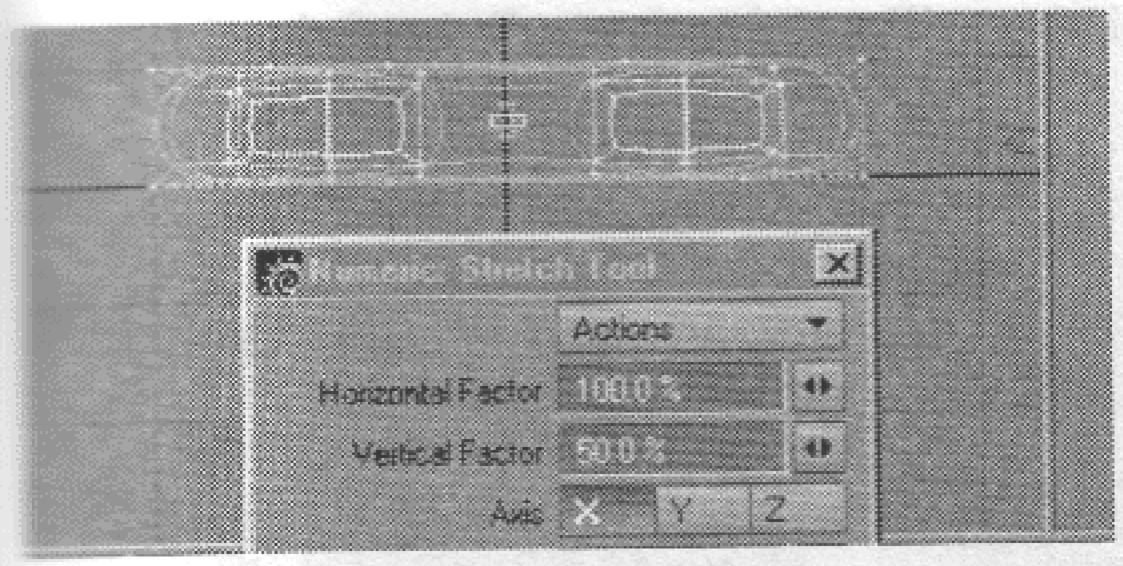
Notice how you can often use either the information display or the Numeric panel for modification feedback.

10 Deselect the center polygon and choose **Multiply** > **Smooth Shift**. Place your mousepointer in the Back viewport and drag to the right for an offset of 40cm.



Numeric: Smooth Shift
Offset 40 cm
Max Smoothing A... 89.5°

Choose the **Stretch** tool again and place your mousepointer on the Right viewport at 50cm on the Y axis. Drag down to stretch your selection 50% Vertical only.



Numeric: Stretch Tool

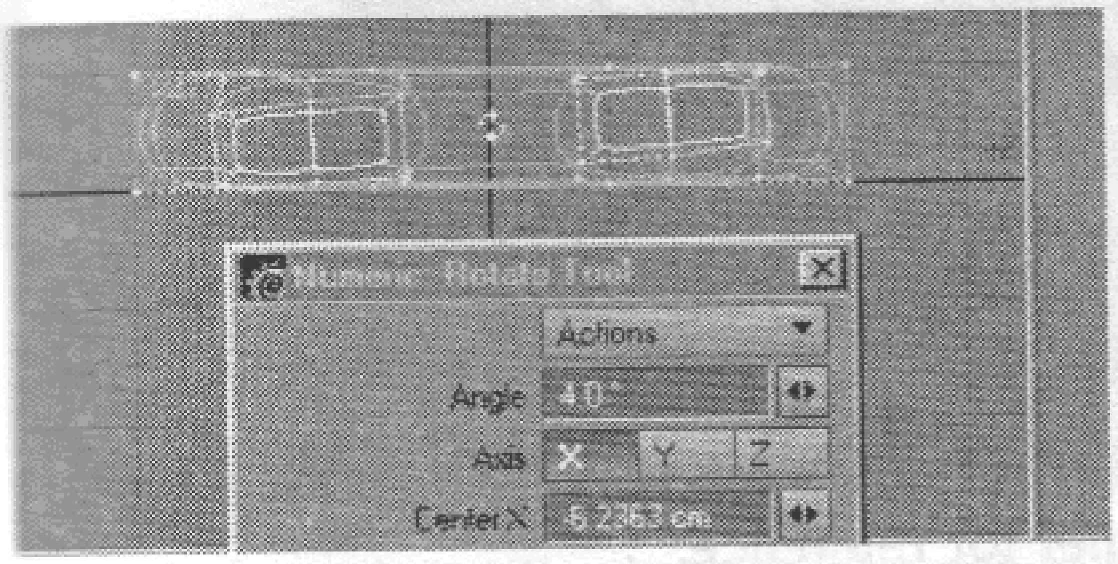
Actions

Horizontal Factor 100.0 %

Vertical Factor 50.0 %

Axis X Y Z

Then, without moving your mouse (it should still be at Y = 50cm), press the Y key to select the **Rotation** tool. Rotate the polygons counterclockwise approximately four degrees.



Numeric: Rotate Tool

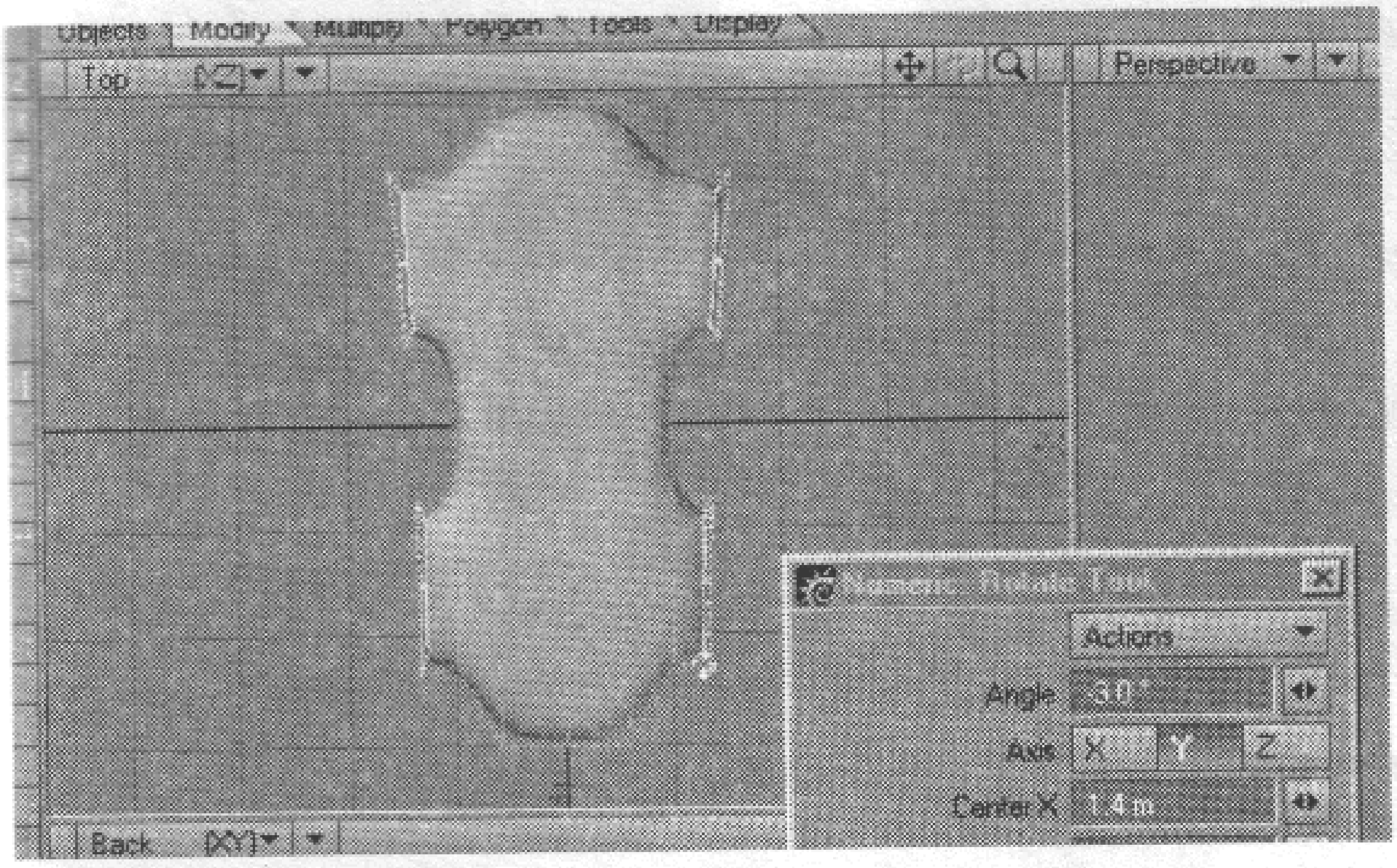
Actions

Angle 4.0°

Axis X Y Z

Center X 6.2363 cm

In the Top viewport, place your rotation cursor on the right-selected polygon's bottom-most point and rotate it -3 degrees. This will angle out the wheel well sides. Note how nice Symmetry works on the opposite side.



Numeric: Rotate Tool

Actions

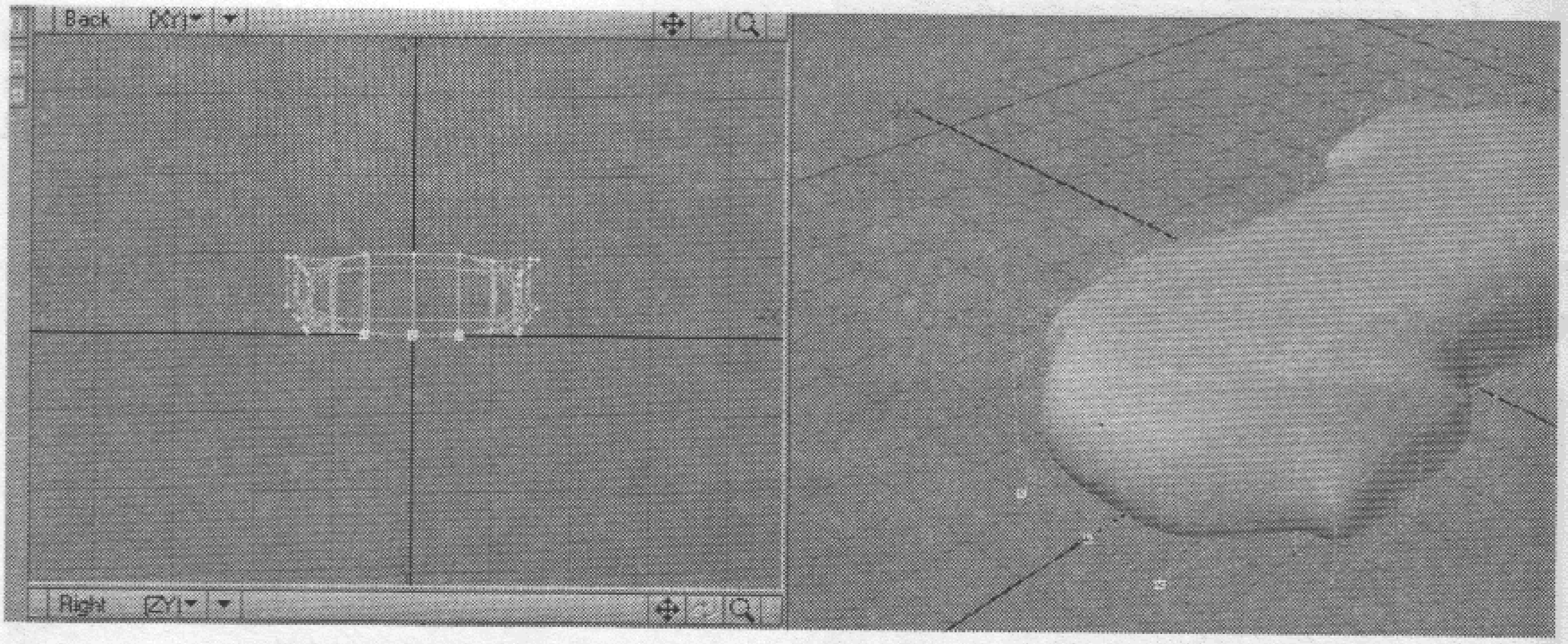
Angle -3.0°

Axis X Y Z

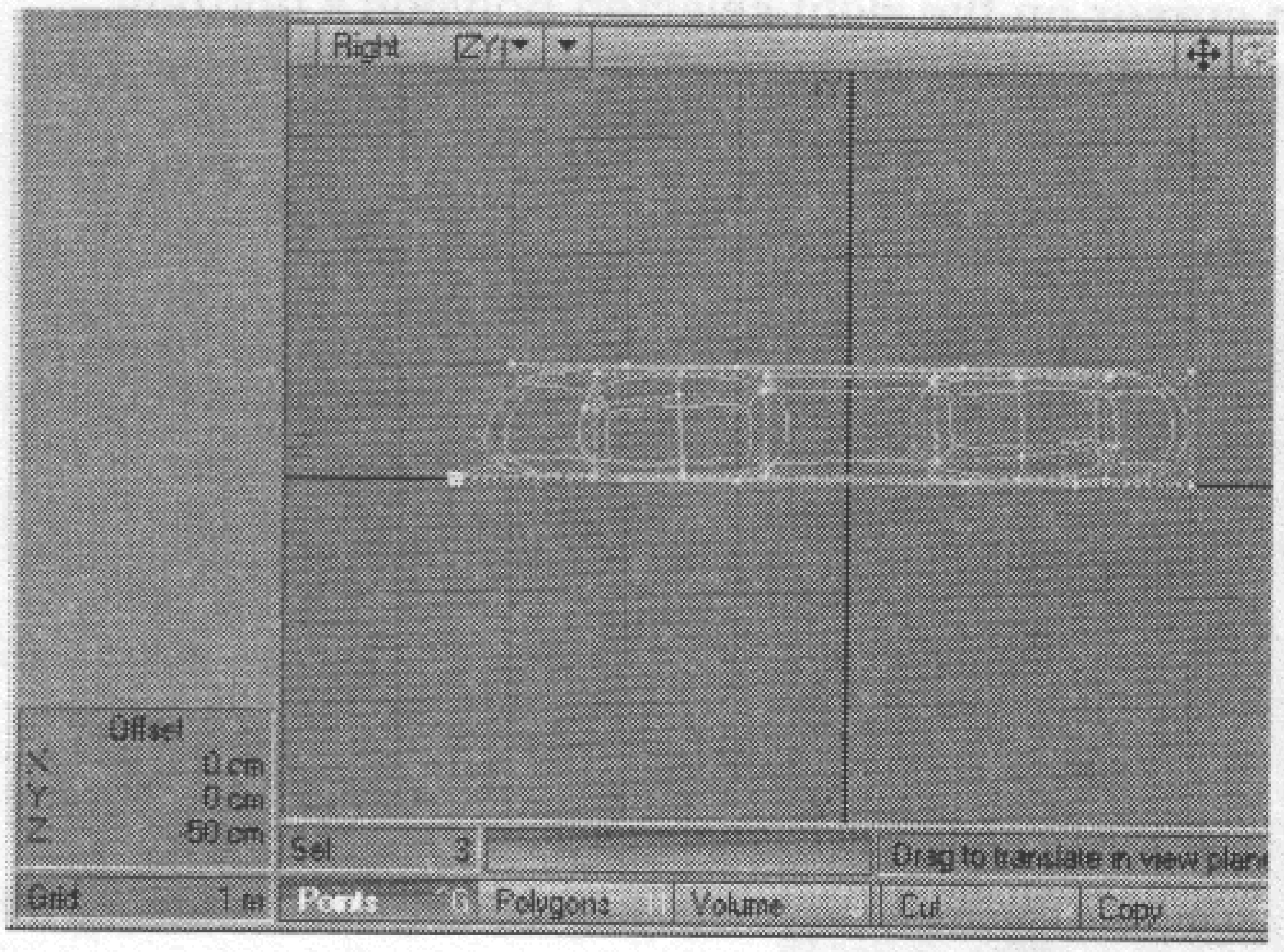
Center X 1.4 m

Pulling out the wheel wells

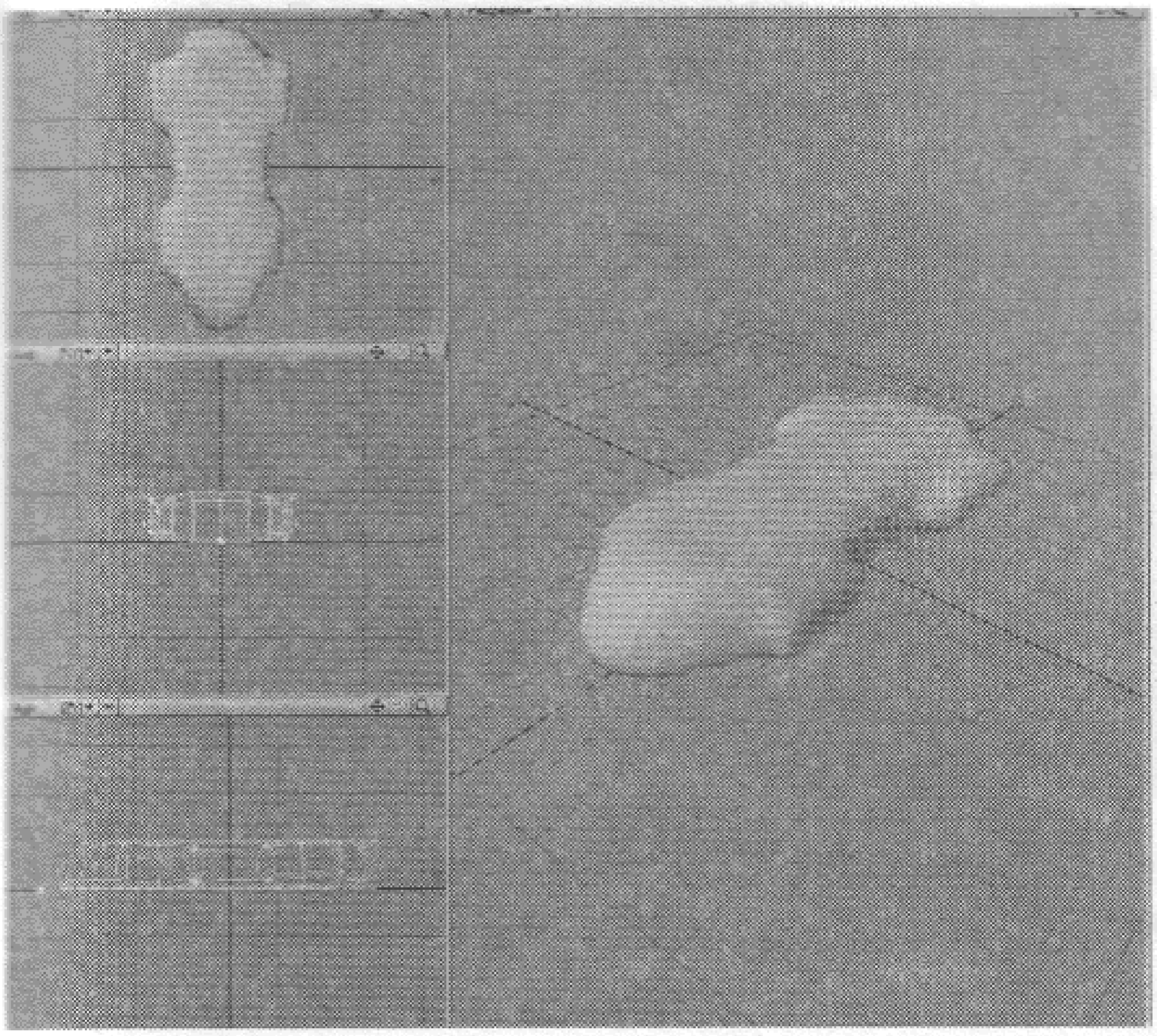
11 Let's start shaping the front of the car. For this we will need to be in **Point** Selection mode (CTRL+G). Select the points along the front's bottom edge, you should end up with three points selected. Your Perspective viewport works well for this.



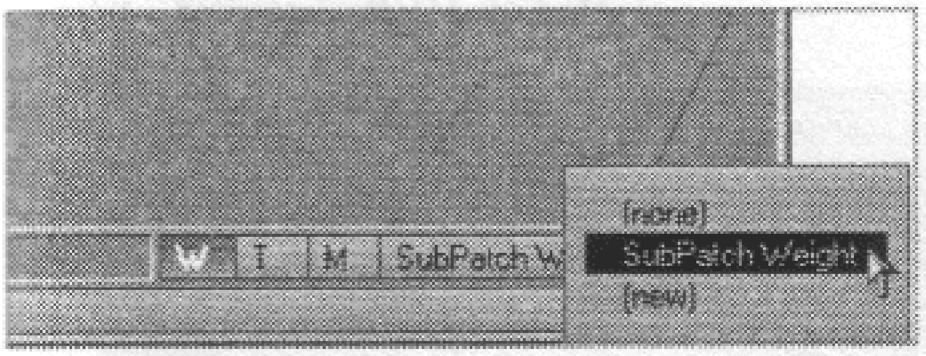
12 Select **Objects** > **Move**. Then, in the Back viewport, drag the points -50cm on the Z-axis. Use the information display in the lower-left corner for reference.


Offset
X 0 cm
Y 0 cm
Z -50 cm
Sel 3
Drag to translate in view plane
Grid 1 m

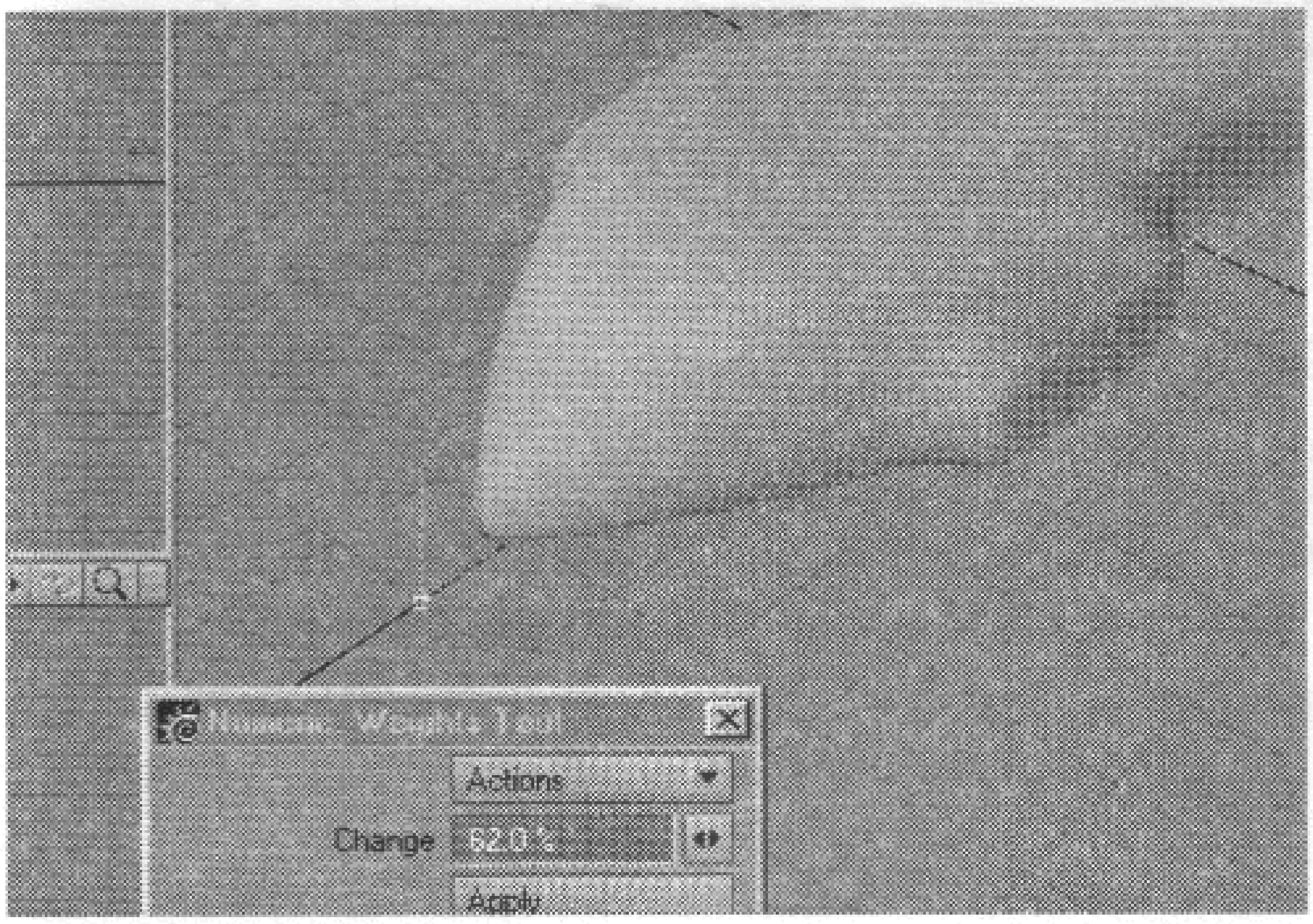
Deactivate the **Move** tool and then deselect the outer two points. (**Symmetry** should still be on, so when you unselect one point it's counterpart across the X-axis will be automatically deselected.) **Move** the remaining selected point out another -50cm on the Z-axis.



13 With the point still selected, let's adjust the contour of the curve using a weight map. Click on the **W** button in the lower-right corner, if it isn't already selected. Then, select **SubPatch Weight** from the pop-up menu to the right of the button.

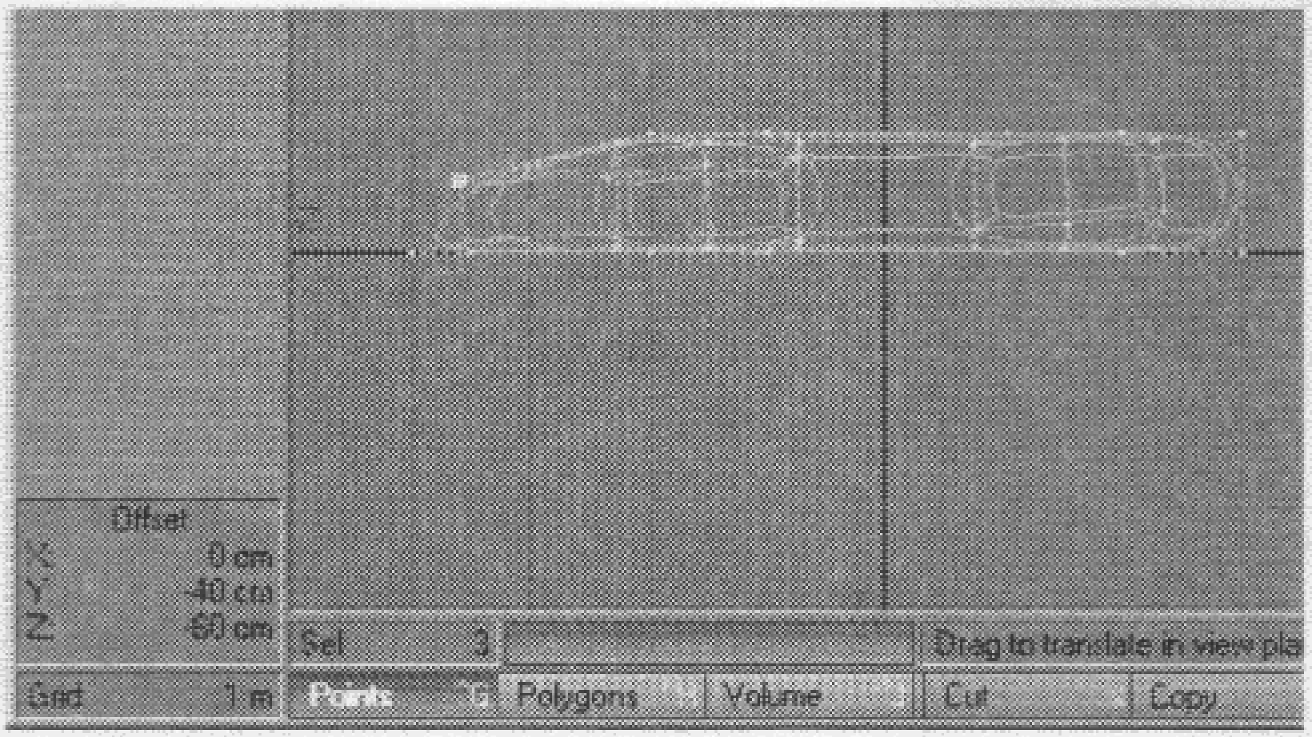


Next, choose **Modify** > **Weights**. Place your mousepointer over the selected point in any viewport. Drag your mouse until the **Change** field on the Numeric panel is at 62.0%. This changes the weighting of this point and increases its effect on the SubPatch surface.



Change 62.0%

14 Drop the **Weight** tool by tapping the SPACEBAR or clicking its button again. Deselect all the points. Select the three points along the front's top edge. In the Right viewport, **Move** these forward and down -40cm Y-axis and -60cm Z-axis.



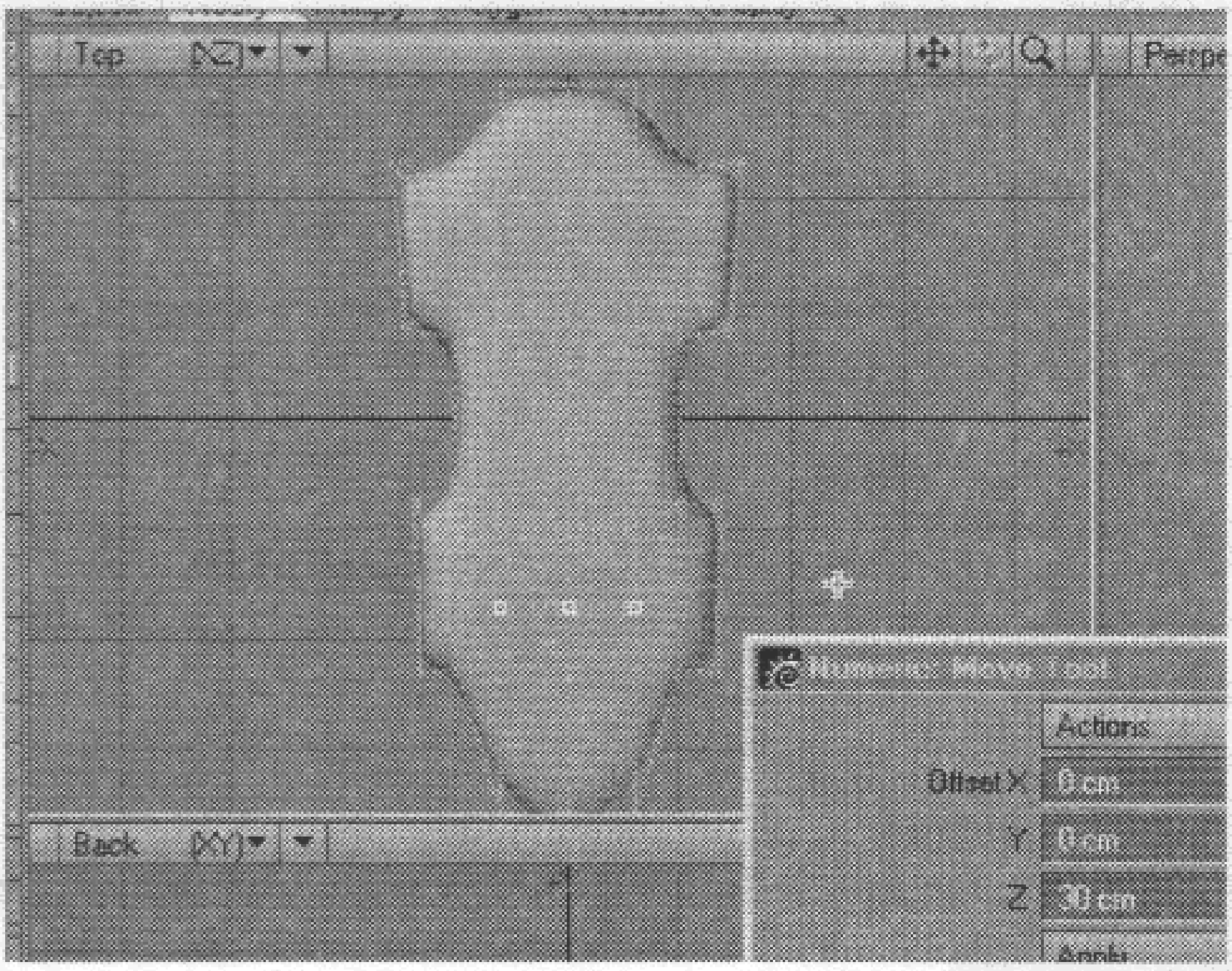
Offset
X 0 cm
Y -40 cm
Z -60 cm

Sel 3

Drag to translate in view pla

15 In the Top viewport, select only the three points at -2m Z on the top of the model and **Move** them back 30cm on the Z axis.

☞ **NOTE**
Remember to drop your tool between selection changes.



Numeric: Move Tool

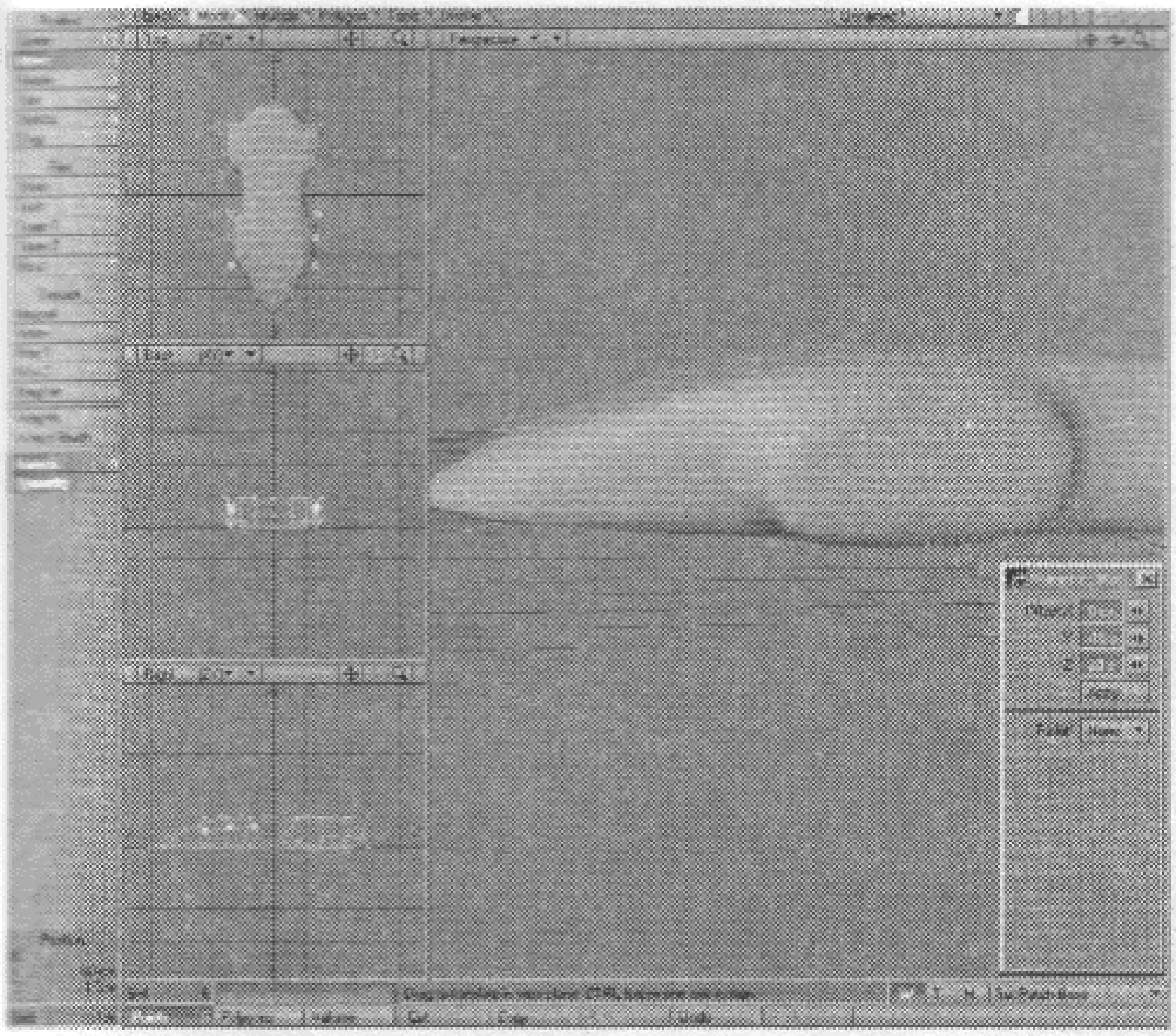
Actions

Offset X 0 cm
Y 0 cm
Z 30 cm

Apply

☞ **NOTE**
Unless otherwise noted, the points described should be the only points selected.

16 Select the top points of the front wheel wells (three on each side) and in the Right viewport **Move** them up to -10cm Y and 20cm Z. You'll probably find selecting in the Perspective viewport the easiest, since you can rotate it to get at the right points.

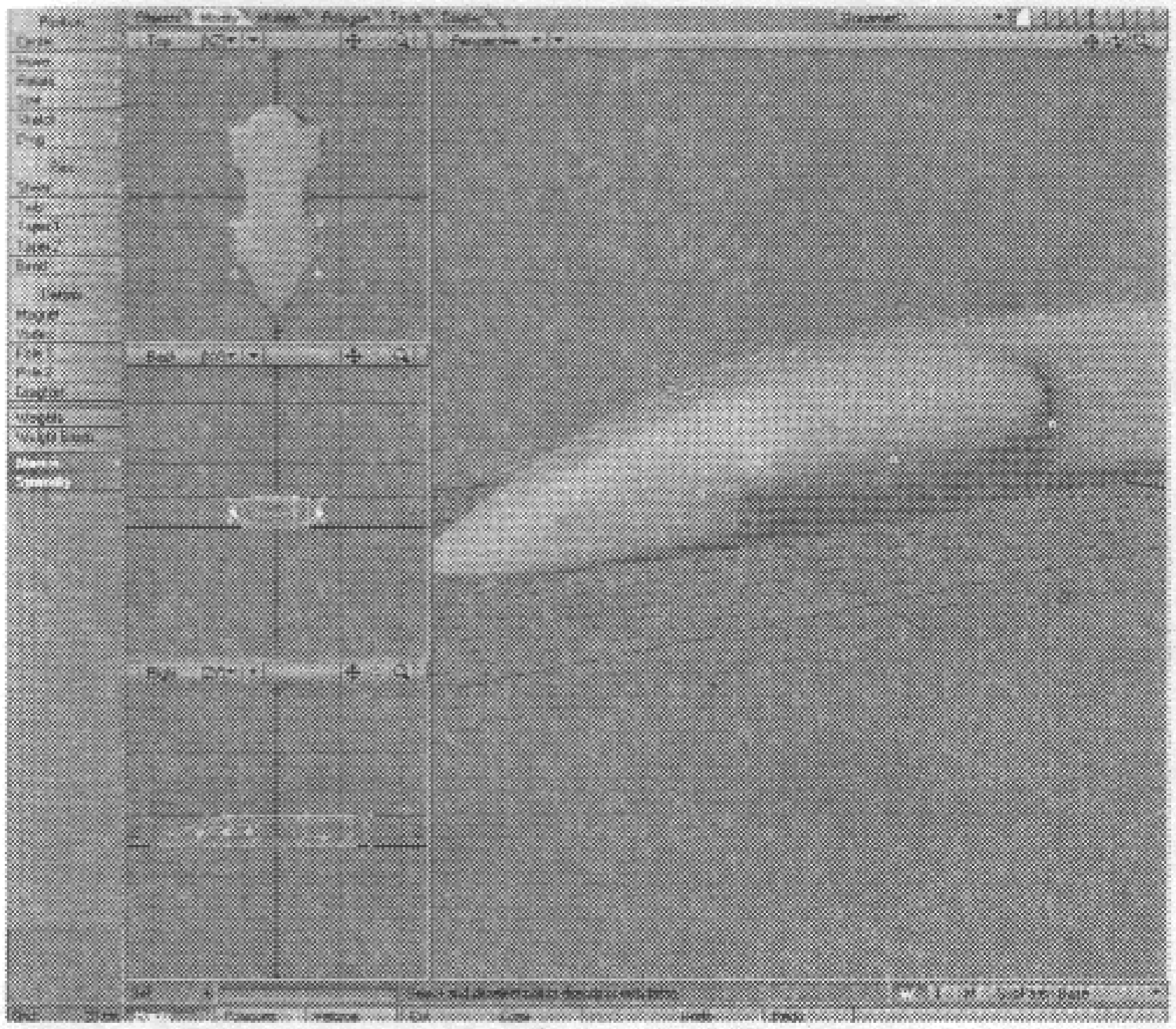


Selecting the top points

☞ **NOTE**

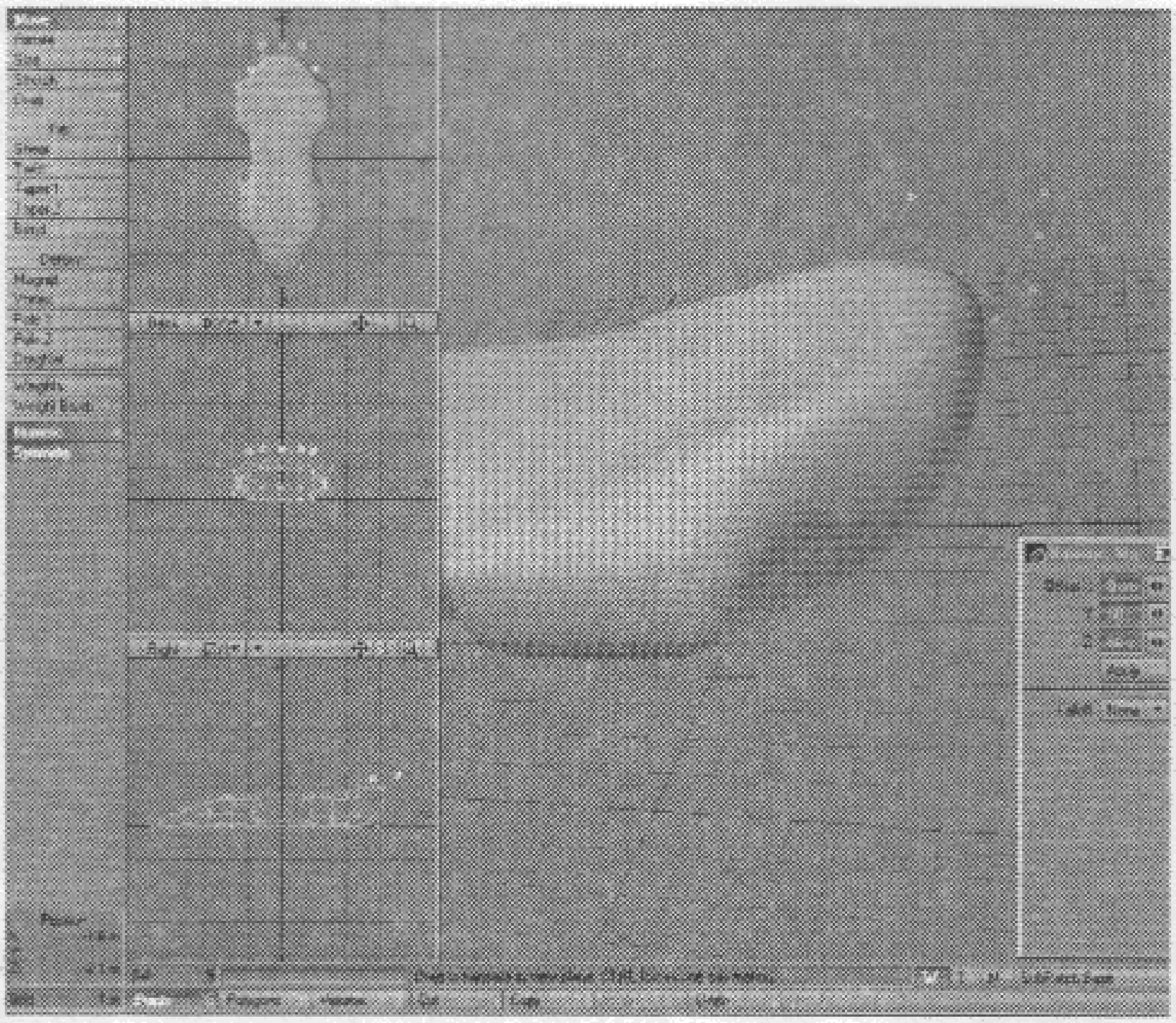
If the cage display bothers you, you can turn them off on the Display Options panel.

17 Select the bottom-outer points of the front wheel well and in the Right viewport **Move** them -10cm Z and 40cm Y.



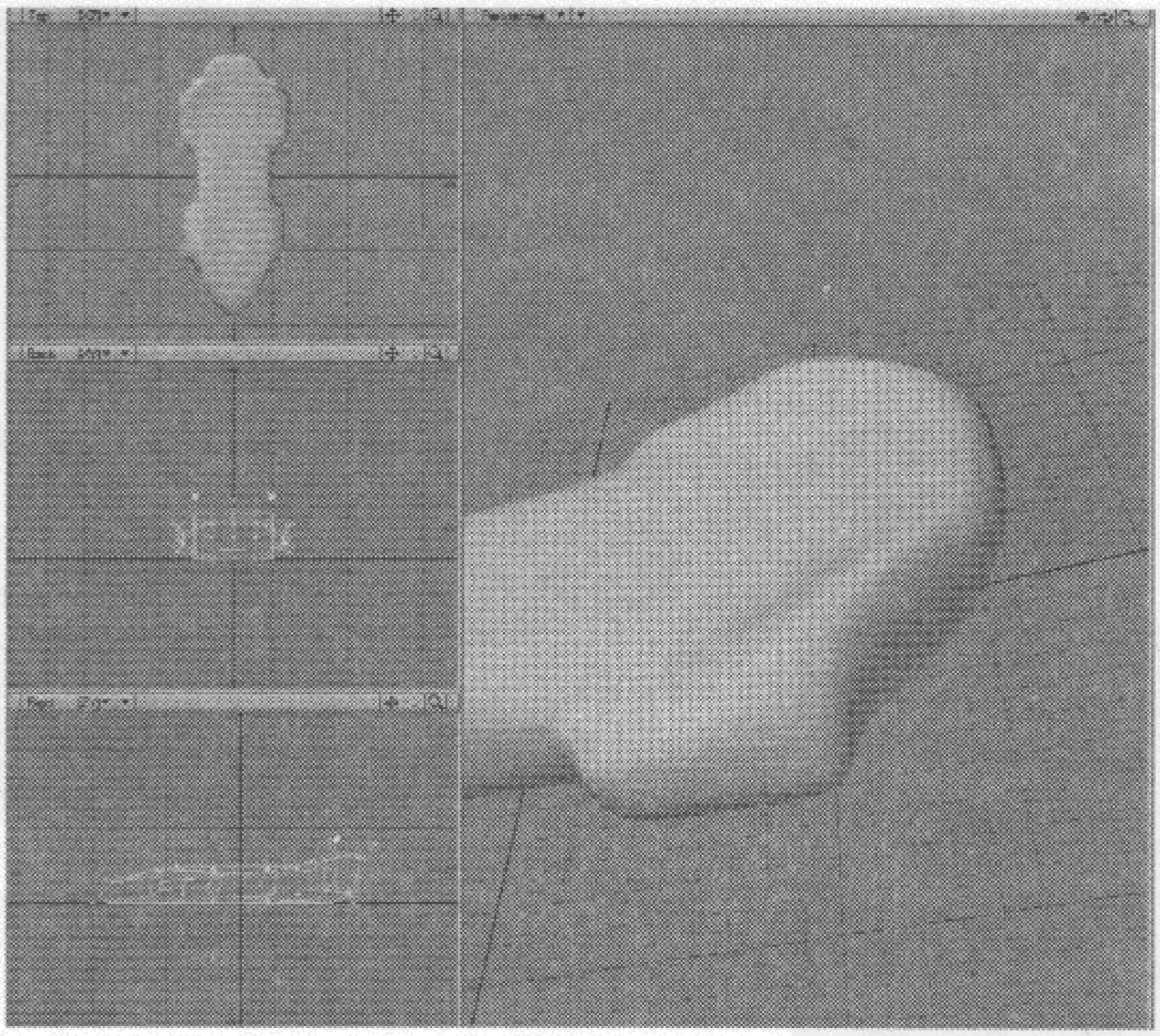
Adjusting the wheel well points

18 Select the five points along the top back (the edge of the trunk area) and in the Right viewport **Move** them 60cm on the Y and Z axes.

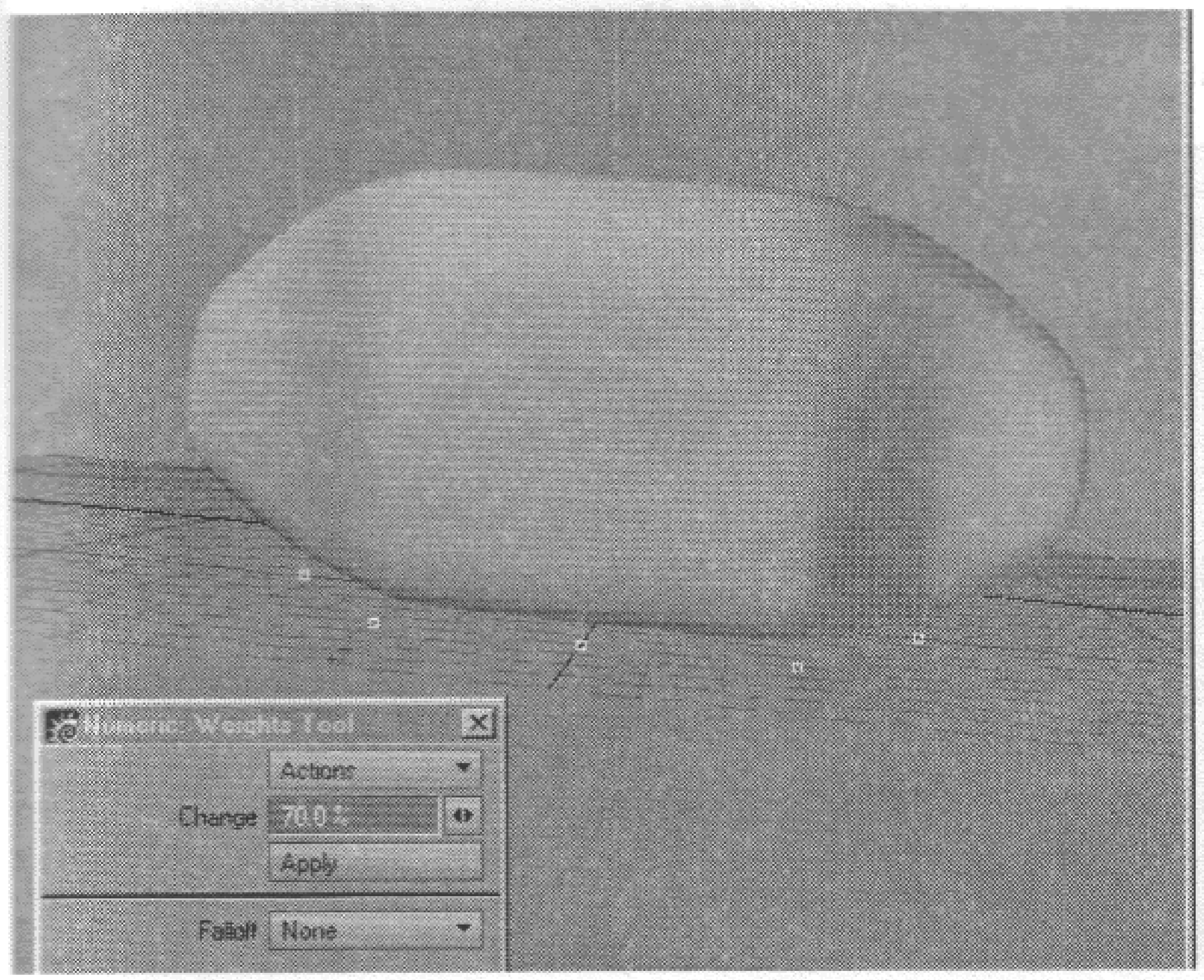


Dragging the trunk out

19 Deselect all the points, except the outer most two, and in the Right viewport **Move** them 20cm Y and -30cm Z.



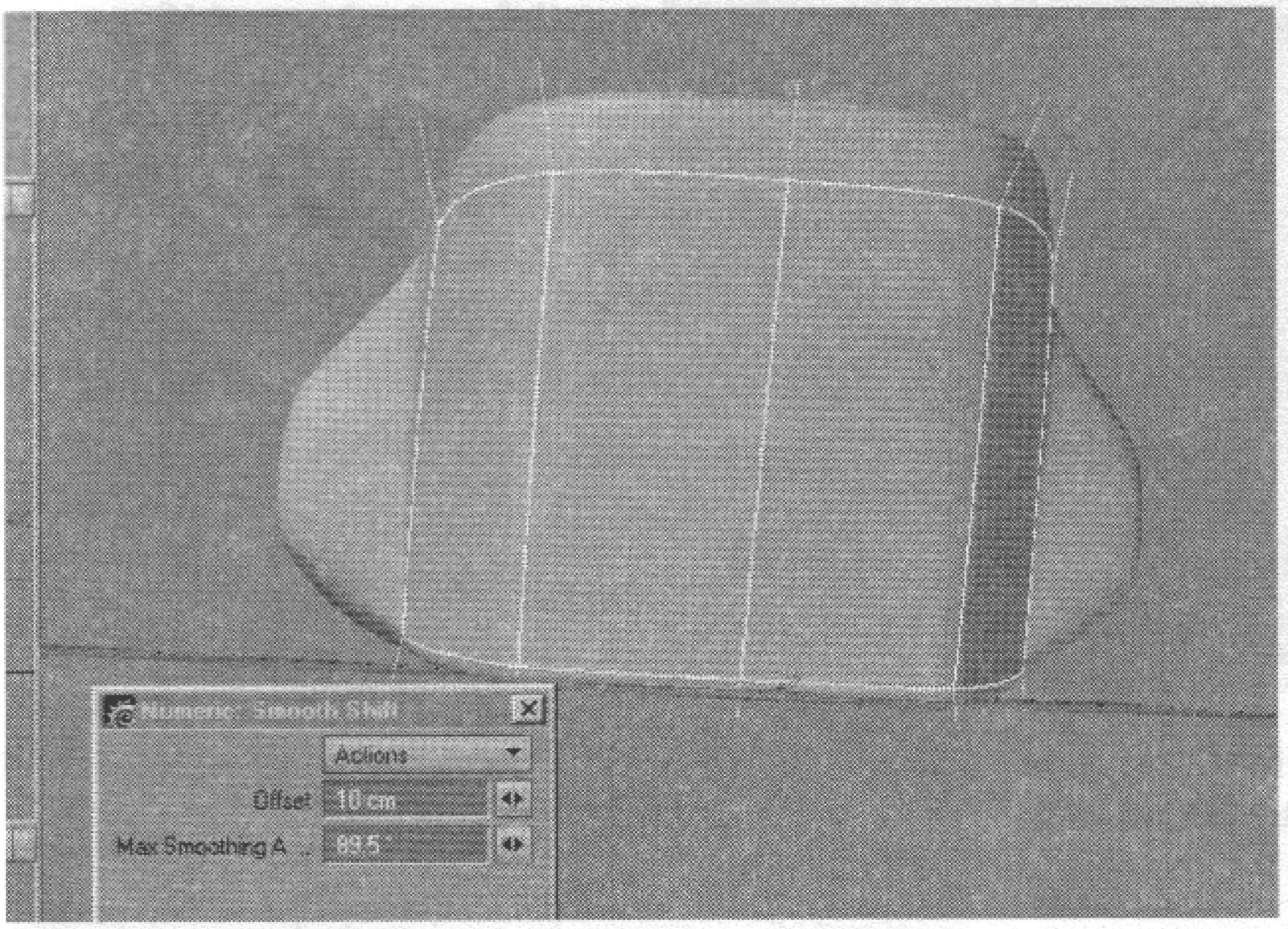
20 Select the five points long the bottom back edge. Select the **Weights** tool again. (**SubPatch Weight** should still be selected as the Weight map.) In the Numeric panel, set the **Falloff** to **None**. This allows you to apply the change evenly to all of the selected points. Drag your mouse until the **Change** field is 70.0%—since **Falloff** is set to **None**, you don't have to have your mousepointer over any point.


Numeric: Weights Tool

Change 70.0 %

Falloff None

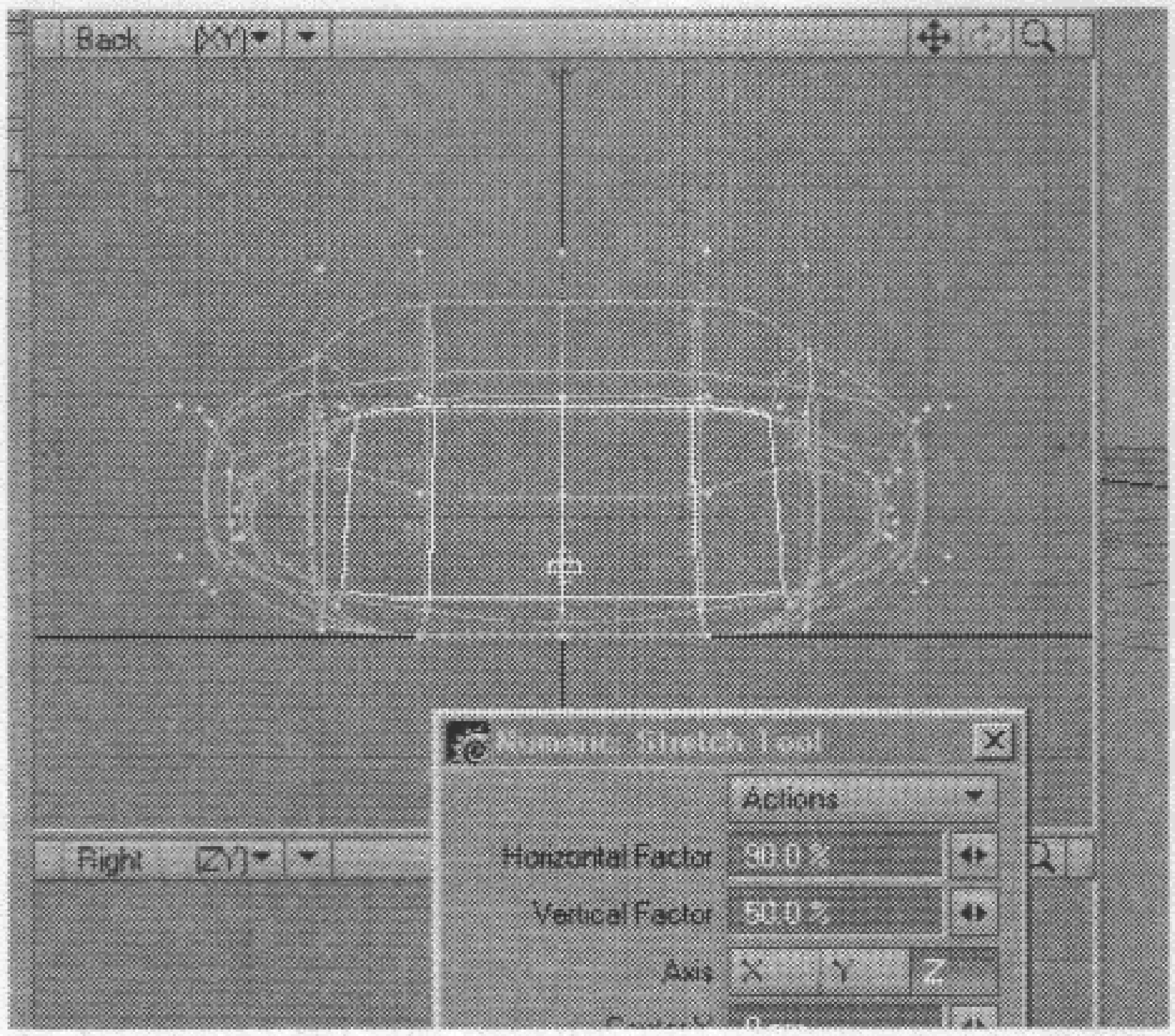
21 Select the back-middle-four polygons. Then, select the **Smooth Shift** tool. In the Numeric panel, select **Activate** from the **Actions** pop-up menu. Enter 10cm in the **Offset** field. This is how you use the Numeric panel for precise values with tools.


Numeric: Smooth Shift

Offset 10 cm

Max Smoothing A... 89.5°

In the Back viewport, **Stretch** the selected polygons 90% Horizontal and 50% Vertical with your mousepointer on Y 30cm.



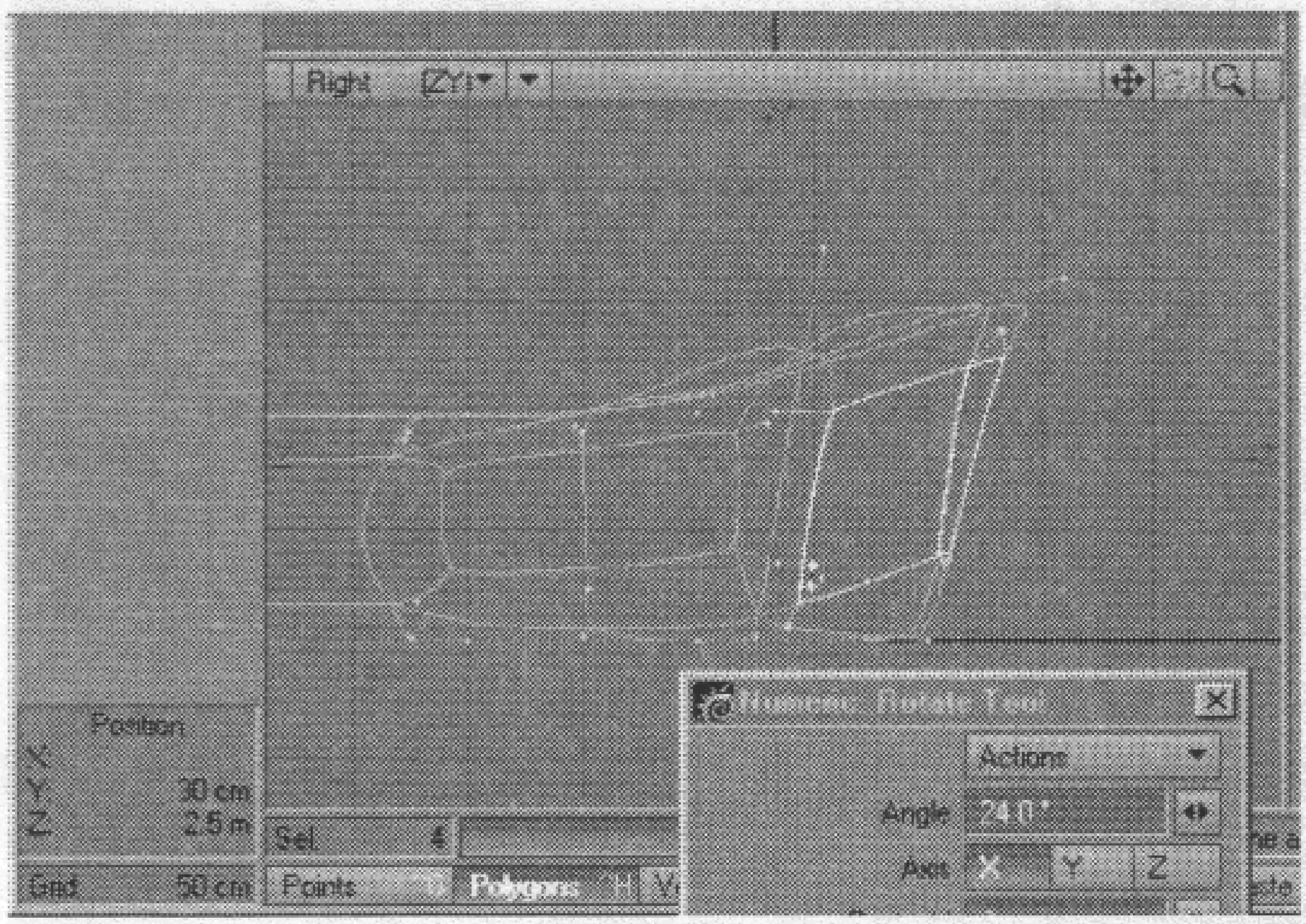
Numeric: Stretch Tool

Horizontal Factor 90.0 %

Vertical Factor 50.0 %

Axis X Y Z

in the Right viewport, with your mousepointer at Y 30cm and Z 2.5m, **Rotate** along the X axis about 24 degrees.



Numeric: Rotate Tool

Angle 24.0°

Axis X Y Z

Position

X:

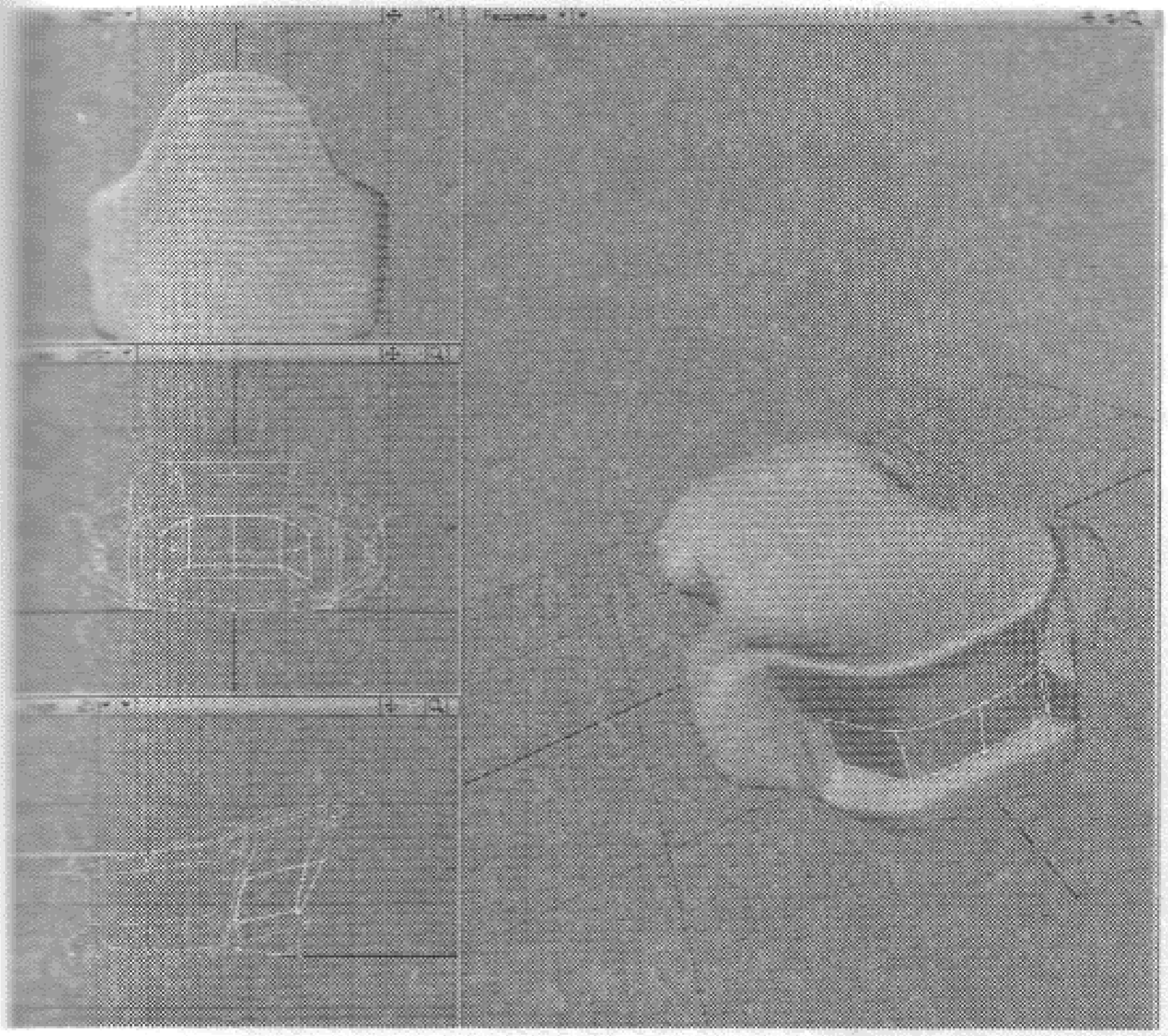
Y: 30 cm

Z: 2.5 m

Grid 50 cm

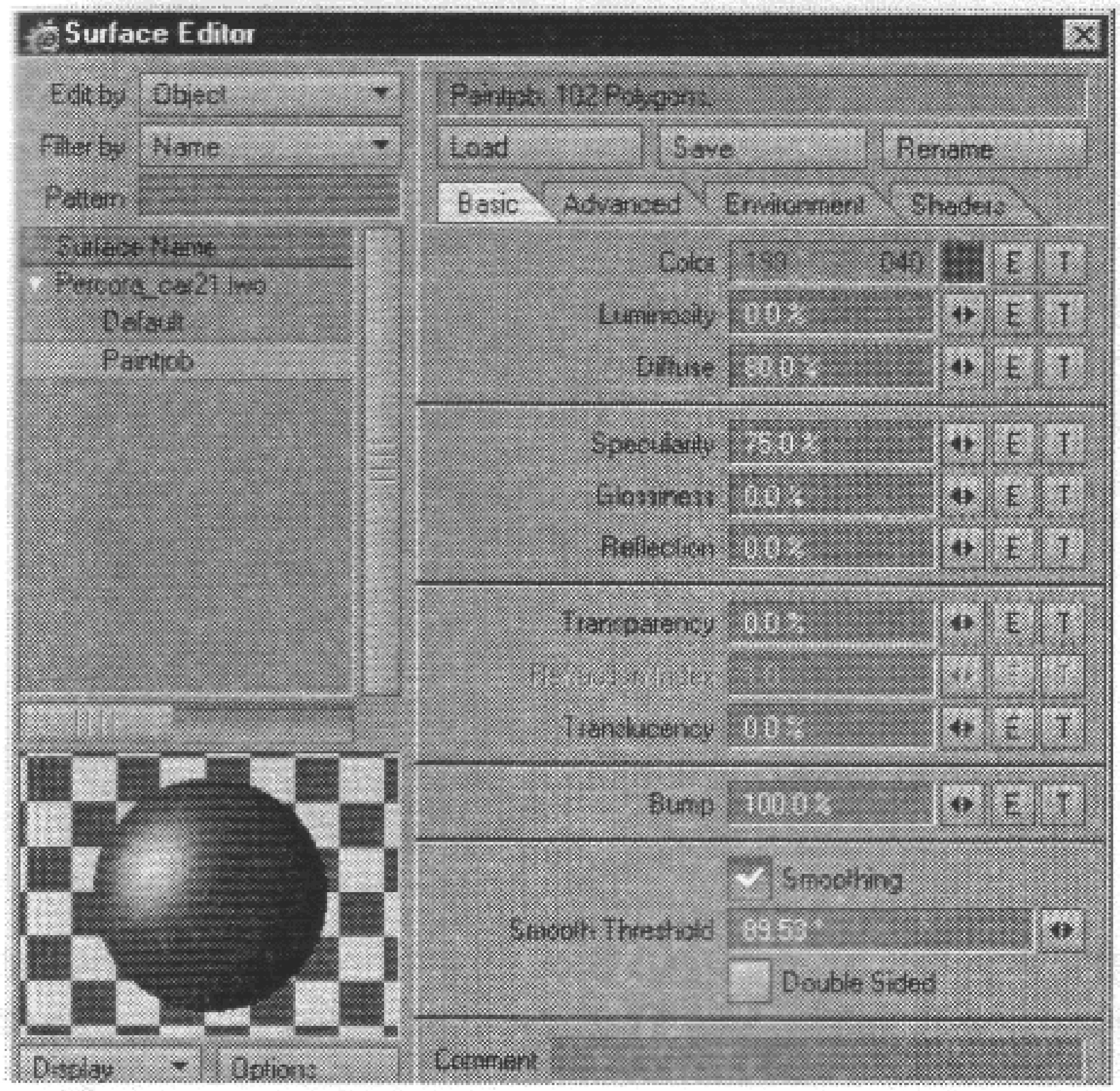
Sel. 4

Smooth Shift the same four polygons using an Offset of 0cm. **Stretch** them 50% Vertical and 80% Horizontal with your cursor at Y 50cm in the Back viewport. **Move** them along the Z -10cm and the Y 10cm.



22 Make sure no polygons are selected. Choose **Polygon** > **Surface** and name the surface PAINTJOB. (When no polygons are selected, it is like all polygons are selected and the surface name is applied to all.)

Choose **Objects** > **Surface Editor** to open the Surface Editor. The Paintjob surface should already be highlighted, but if it is not, select it. Change the color to red 199, green 0, and blue 040 by dragging each RGB color number to the left. Set the **Diffuse** value to 80%, **Specularity** to 75%, **Glossiness** to 35%, and **Reflectivity** to 80%. **Smoothing** should be checked.



Surface Editor

Edit by: Object

Filter by: Name

Pattern:

Surface Name
- Percora_car21.lwo
  - Default
  - Paintjob

Paintjob: 102 Polygons

Color: 199 040

Luminosity: 0.0 %

Diffuse: 80.0 %

Specularity: 75.0 %

Glossiness: 0.0 %

Reflection: 0.0 %

Transparency: 0.0 %

Translucency: 0.0 %

Bump: 100.0 %

☑ Smoothing

Smooth Threshold: 89.53°

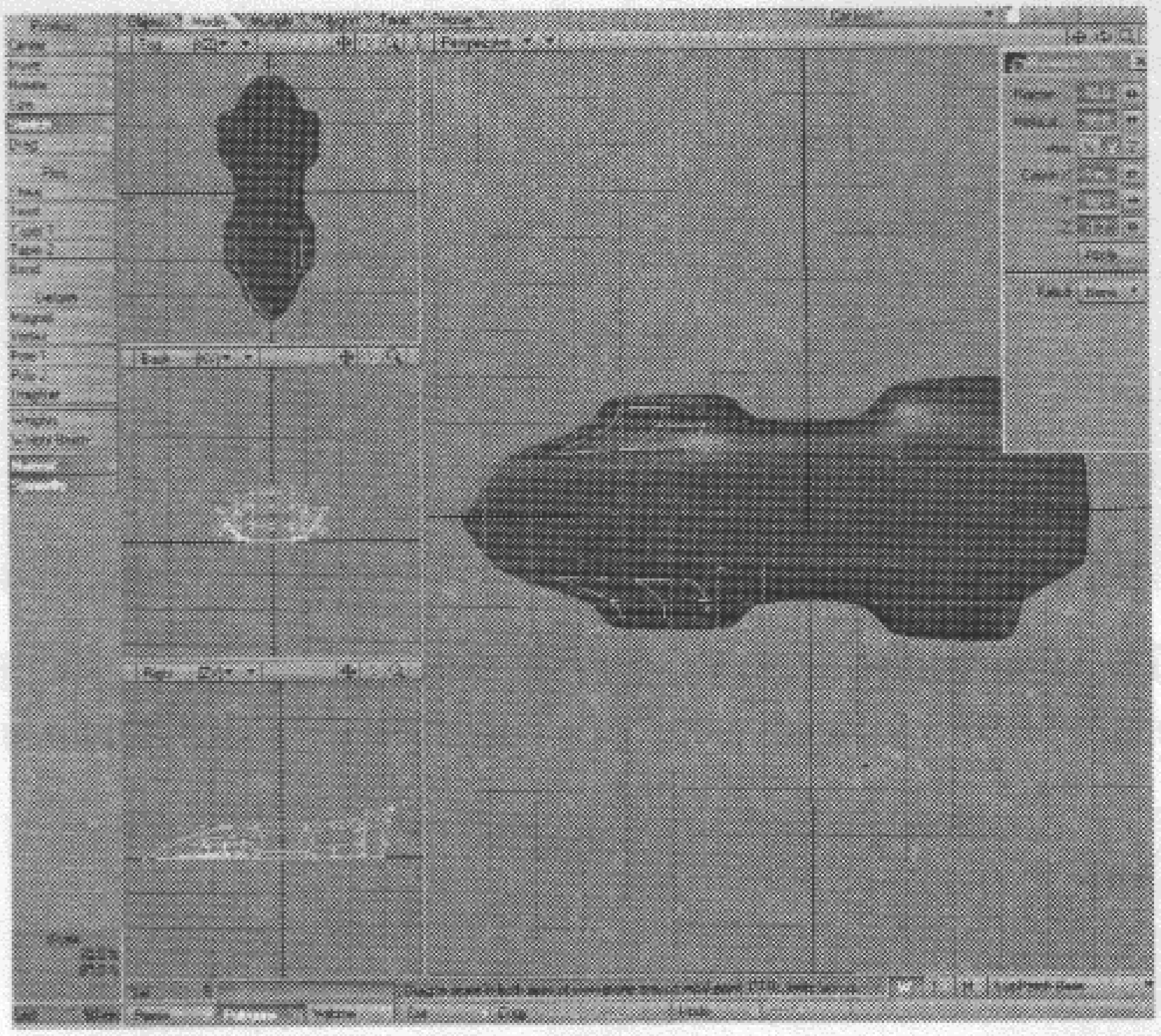
□ Double Sided

Comment:

23 Rotate the car in the perspective window so that we can easily see the bottom. Make sure **Symmetry** is still active.

24 Select the polygons under the front wheel wells; four on each side. **Smooth Shift** them -10cm with a **Max Smoothing Angle** of 180. (This **Max Smoothing Angle** will make sure we affect the greatest number of polygons.)

In the Top viewport, place your mousepointer in the center of the right set of four polygons. **Stretch** them 80% Horizontal and 70% Vertical.

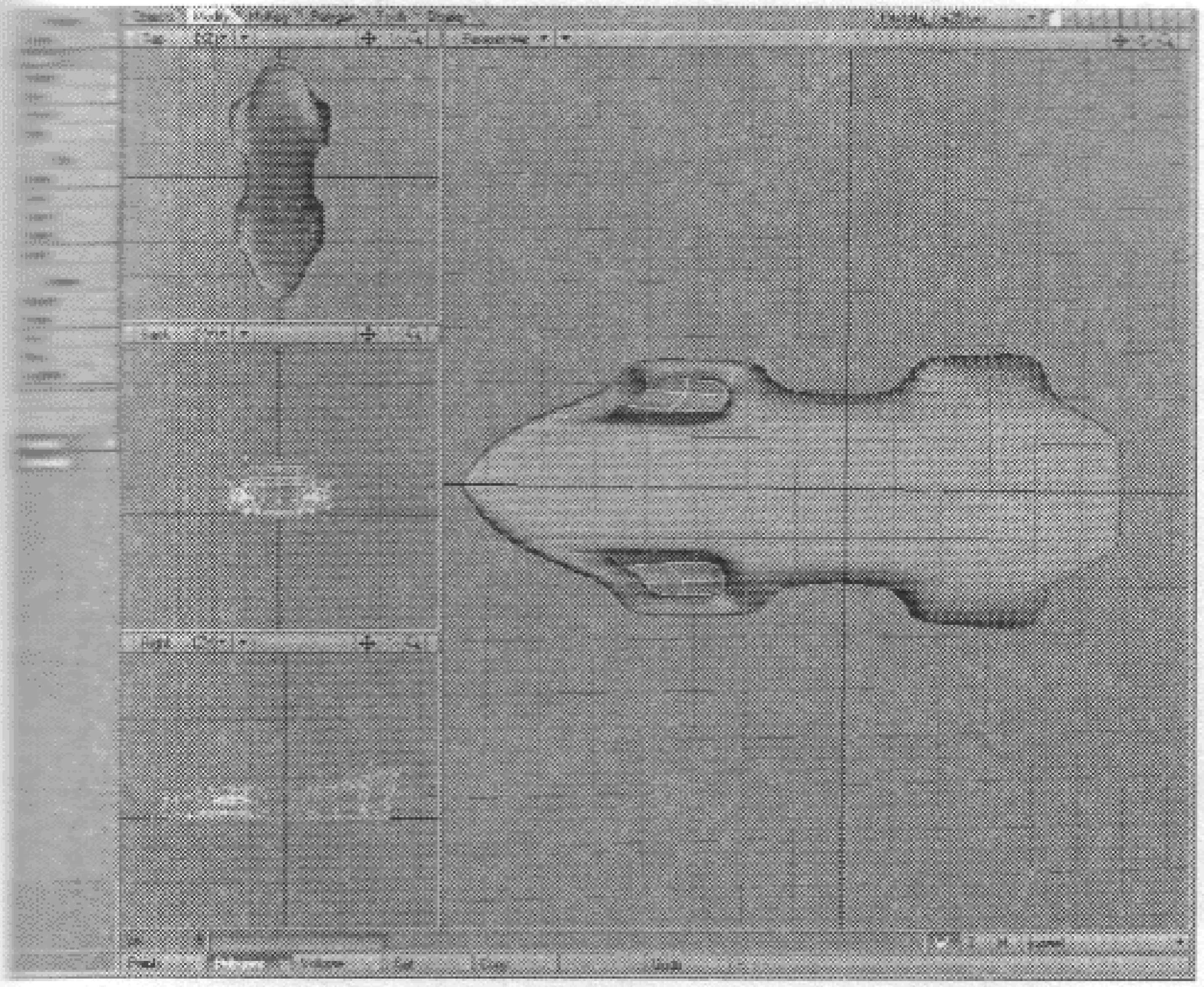


Smooth Shifting on the fly

25 Smooth Shift again -10cm with a **Max Smoothing Angle** of 180. Since this is the same as the previous **Smooth Shift** operation, you can simply select the tool and then select **Activate** from the **Actions** pop-up menu on the Numeric panel.

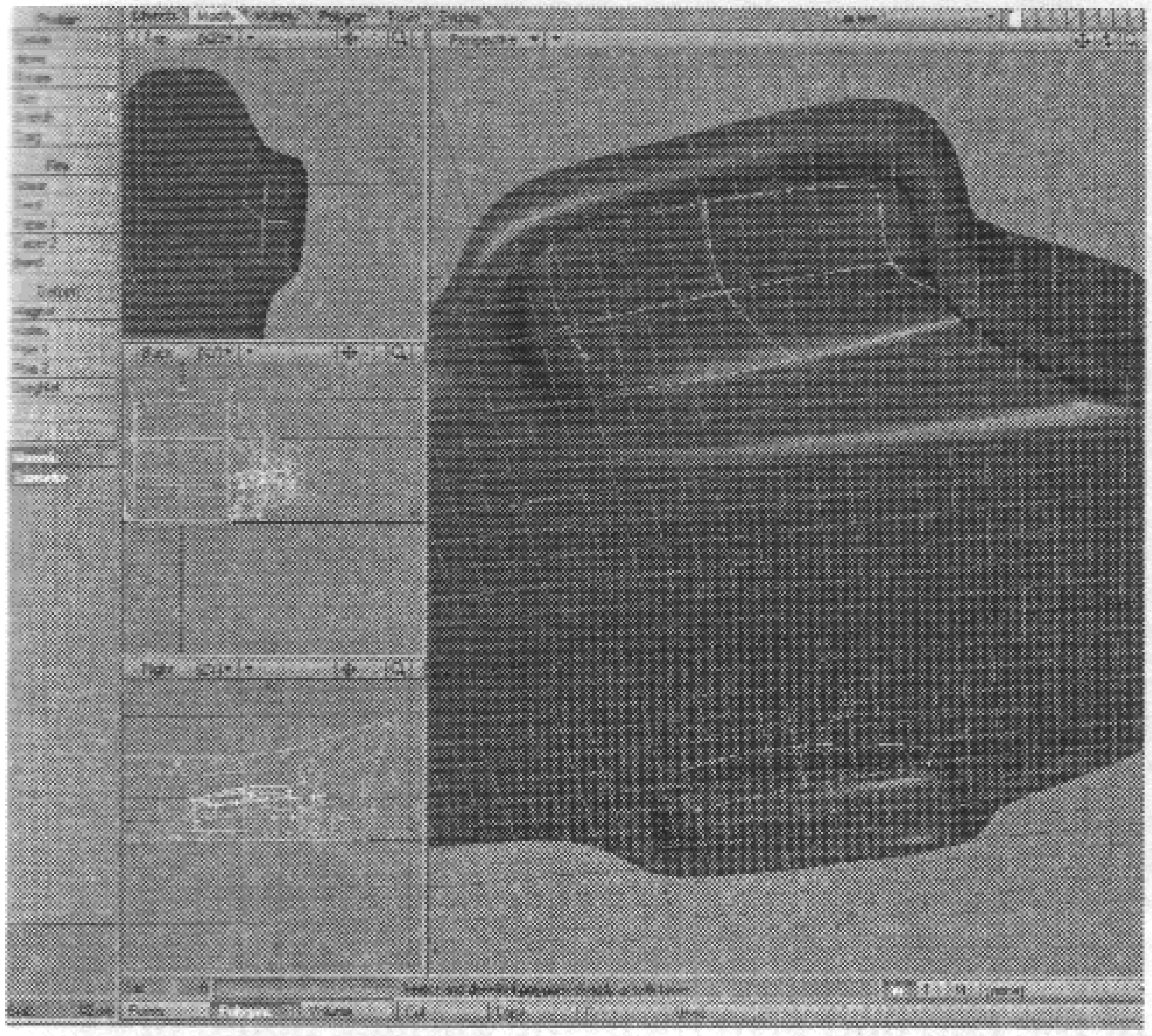
Next, in the Top viewport, **Stretch** 90% Horizontal and 70% Vertical. **Smooth Shift** -40cm with a **Max Smoothing Angle** of 180.

Now, in the Back viewport, place you mousepointer at X 70cm and Y 60cm. **Stretch** them 60% Horizontal and 50% Vertical. Then choose the **Move** tool, hold CTRL and drag right 30cm on the X-axis. (Holding the CTRL key constrains the axis to the initial mouse direction.)



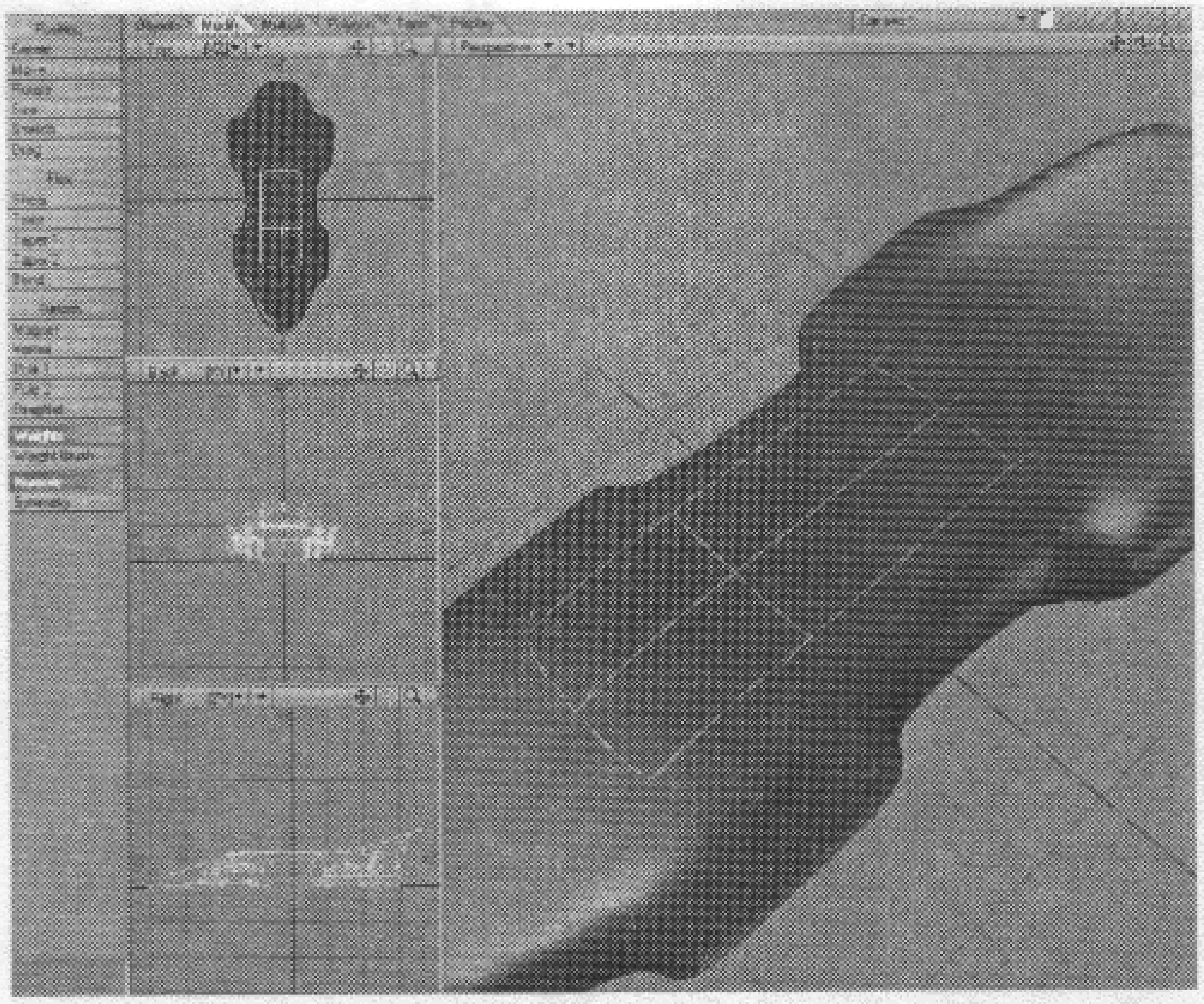
26 Deselect the current polygons, and select the polygons under the back wheel wells; four on each side.

27 **Smooth Shift** the polygons -10cm. In the Top viewport with your mousepointer at the center of the right polygon quad, **Stretch** them 74% Horizontal and 84% Vertical. **Smooth Shift** -20cm with a **Max Smoothing Angle** of 180 and **Move** 10cm X-axis. **Smooth Shift** again for -30cm with a **Max Smoothing Angle** of 180 and **Stretch** 65% Horizontal, 75% Vertical in the Top viewport.



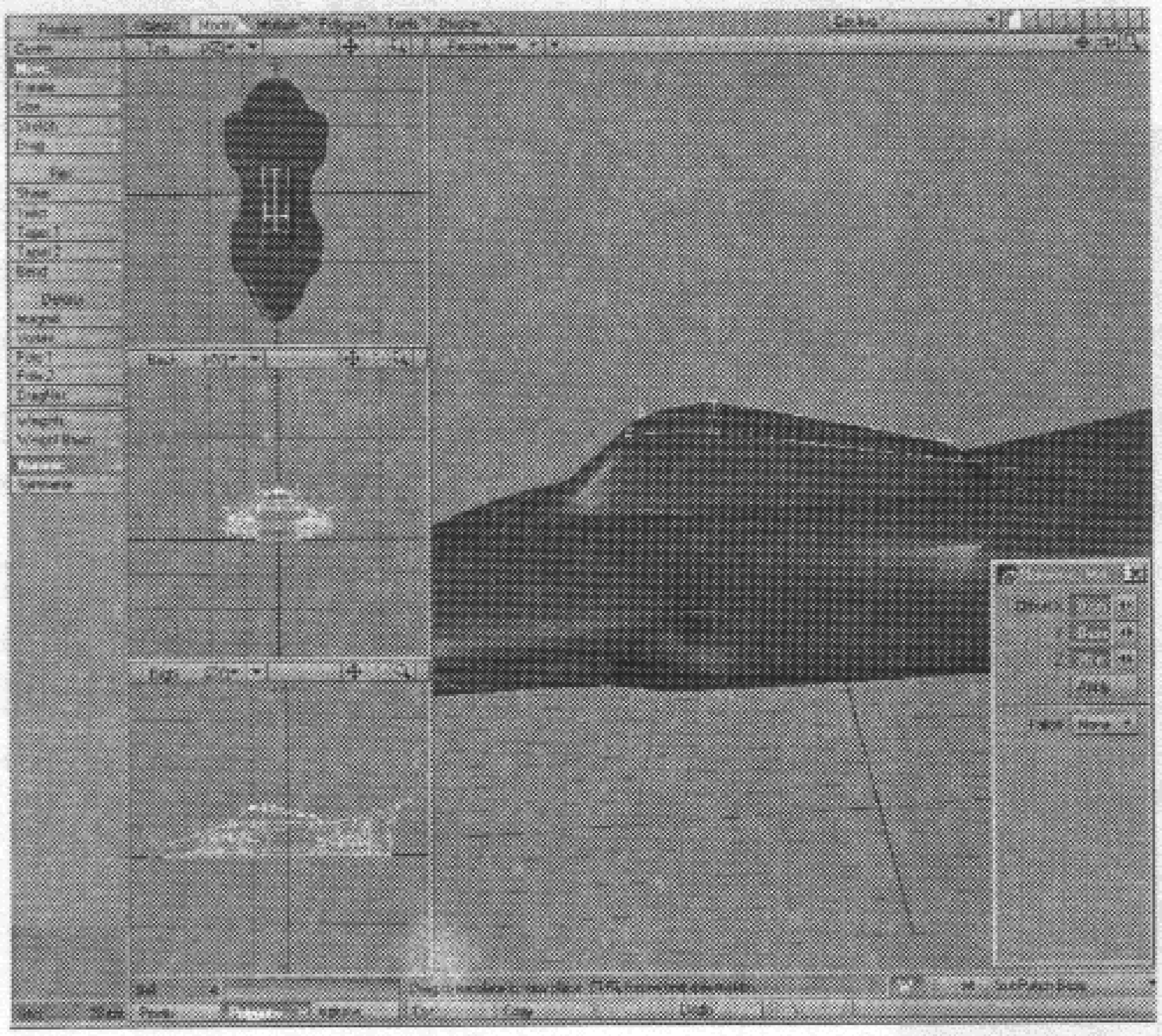
Adjusting the top of the rear wheel wells

28 Drop the current tool and deselect all polygons. Now we need to create the cab. Select the four polygons on the top of the model between the wheel wells.



Selecting the roof polygons

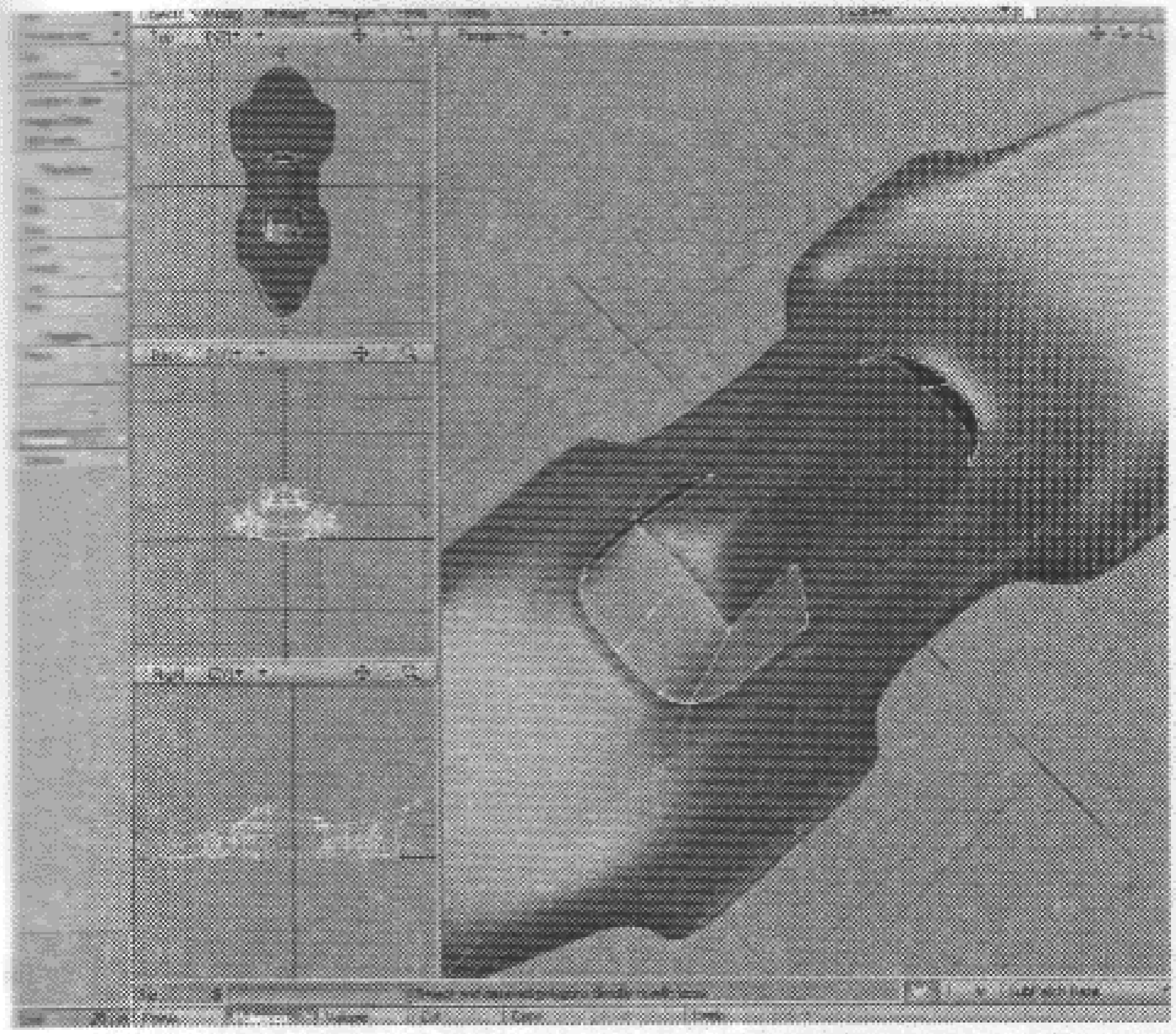
29 **Smooth Shift** -4cm with a **Max Smoothing Angle** of 180. Click your RMB over a viewport to apply the operation and then **Smooth Shift** again 50cm. In the Right viewport, place your mousepointer at Y 1.3m, Z 0m and **Rotate** them -7 degrees. In the Back viewport with your cursor at the Origin (i.e., X 0 and Y 0), **Stretch** these 125% Horizontal and 90% Vertical. **Move** the selected polygons 20cm on the Z axis. Then, deselect the front two polygons and **Move** the back two polygons 20cm more.



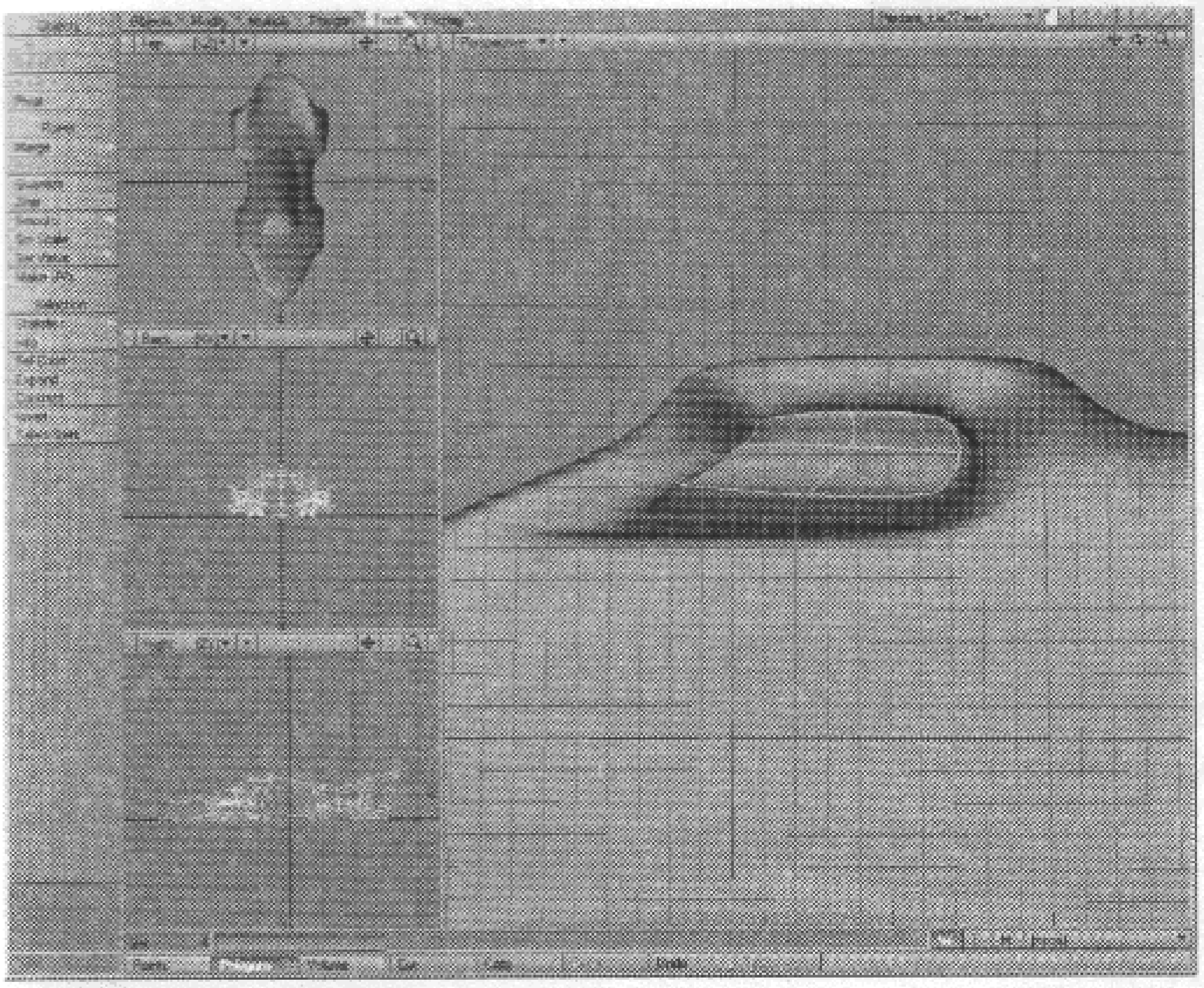
The roof of the love machine takes shape

29b Select the four polygons at the front of the cab and give them the surface name of WINDOW.

Go to the Surface Editor and set the attributes for the windows surface to the following: **Color**: red 095, green 079, blue 174. **Diffuse** 50%, **Specularity** 78%, **Glossiness** 10% and **Reflectivity** 30%. (Yeah, I know windows are generally transparent, but we have no interior!) Now, select the two polygons on the back of the cab and assign the surface Windows to them.

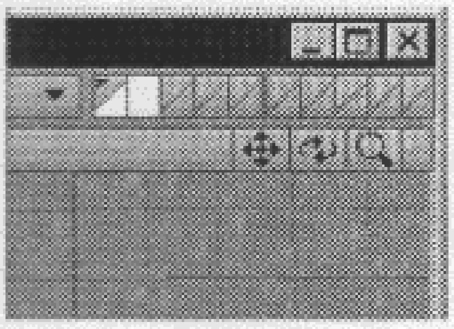


30 It is time now to build some tires. First we need to set ourselves up with a wheel well for reference. In the Perspective viewport, turn your car upside down and zoom in on the left-front wheel well. Turn off **Symmetry**. Select the four polygons inside the wheel well.

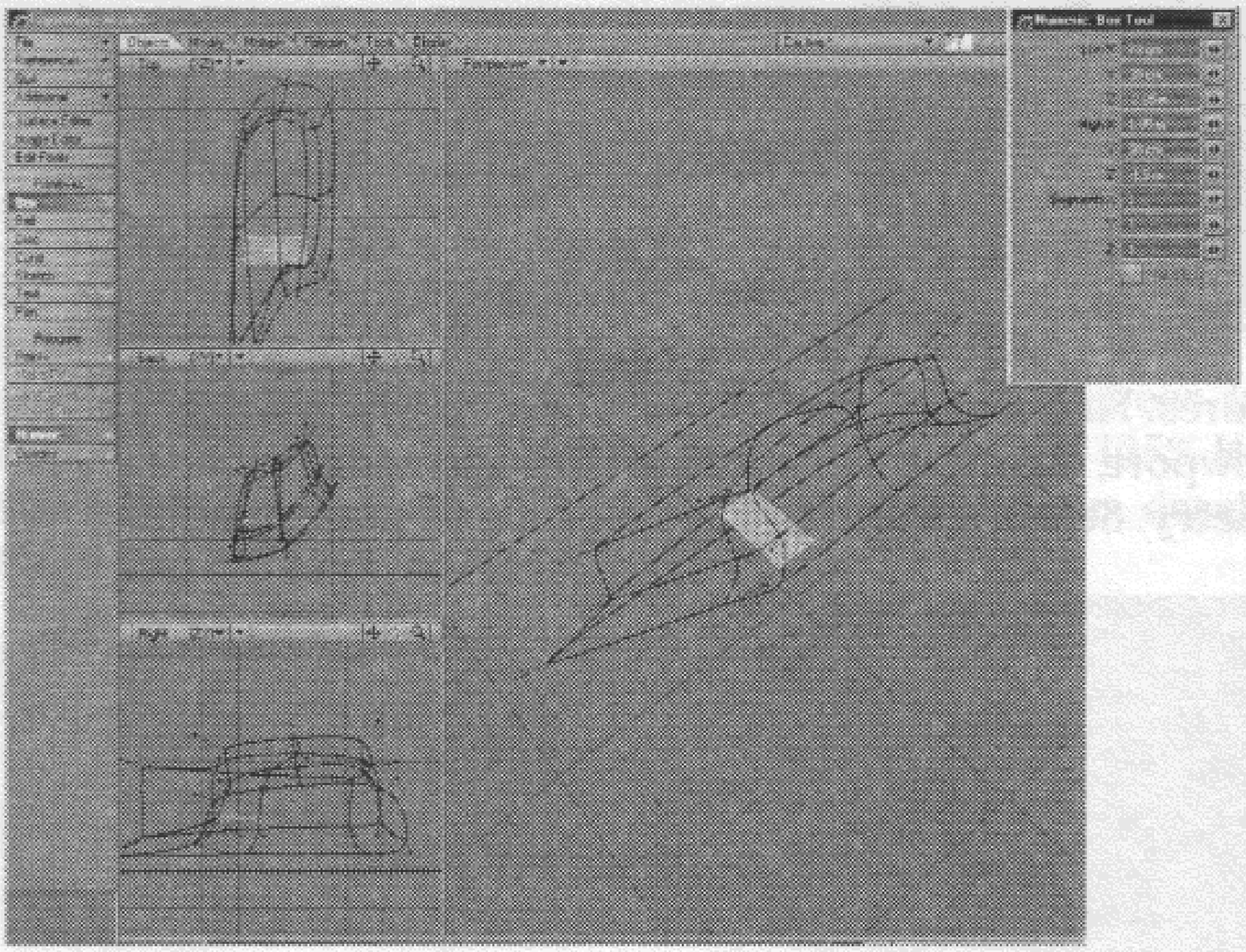


Next, choose **Tools** > **Expand** twice. This command selects contiguous polygons. (It is a step-by-step version of the **Select Connected** command.) Now hide the unselected polygons by choosing **Display** > **Hide Unsel**.

31 Hold the SHIFT key down and click on the bottom of the layer 2 button to put layer 2 in the background. Now, swap foreground and background layer states by choosing **Objects** > **Additional** > **Swap Front and Back Layers**, which should be mapped to your apostrophe (') key. (Of course, you could have set the foreground/background states directly too.)



32 Create a **Box** with the following dimensions: Low XYZ = 72cm, 30cm, -2.05m and High XYZ = 1.12m, 30cm, -1.90m.

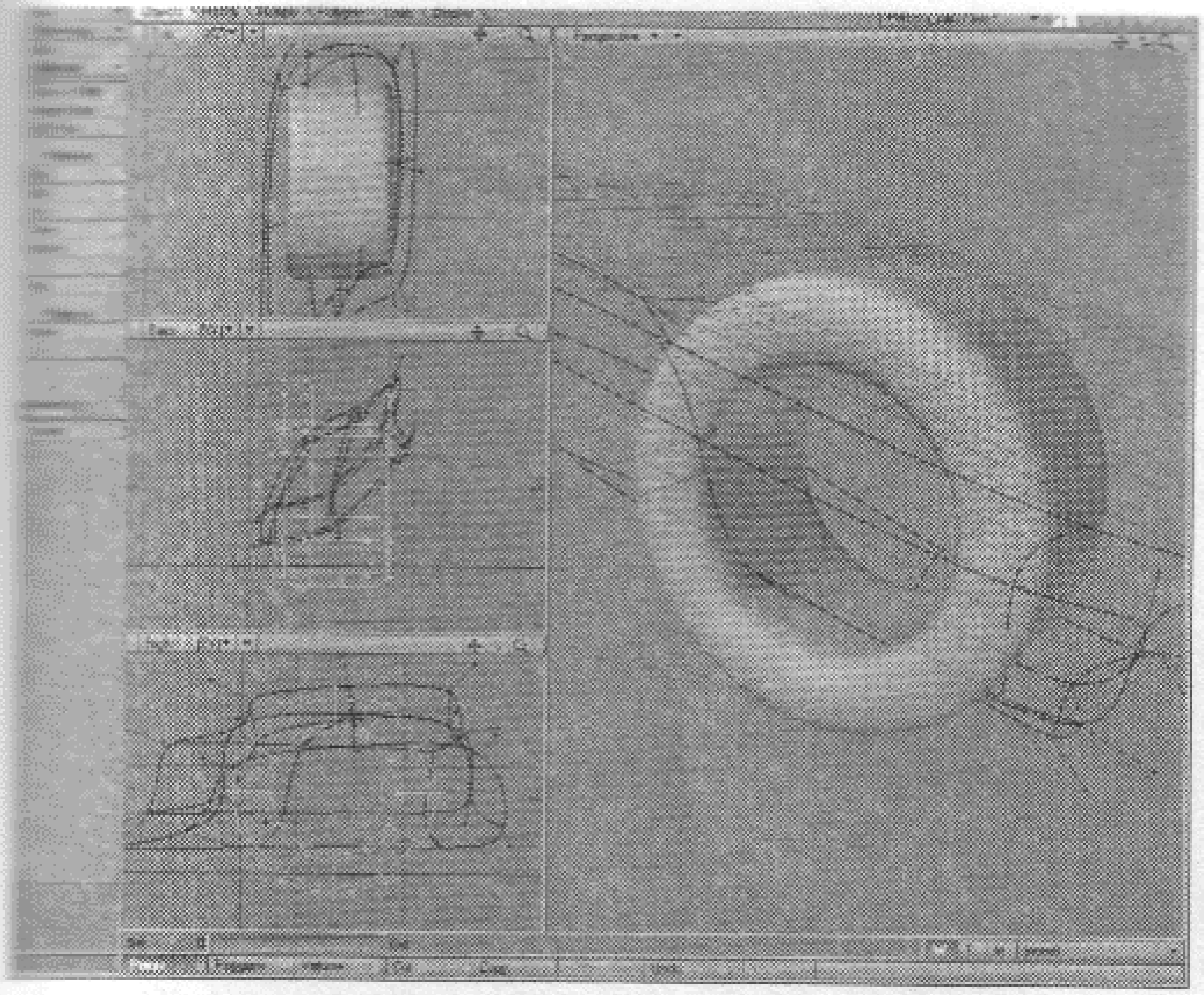


Making a tire from a box.

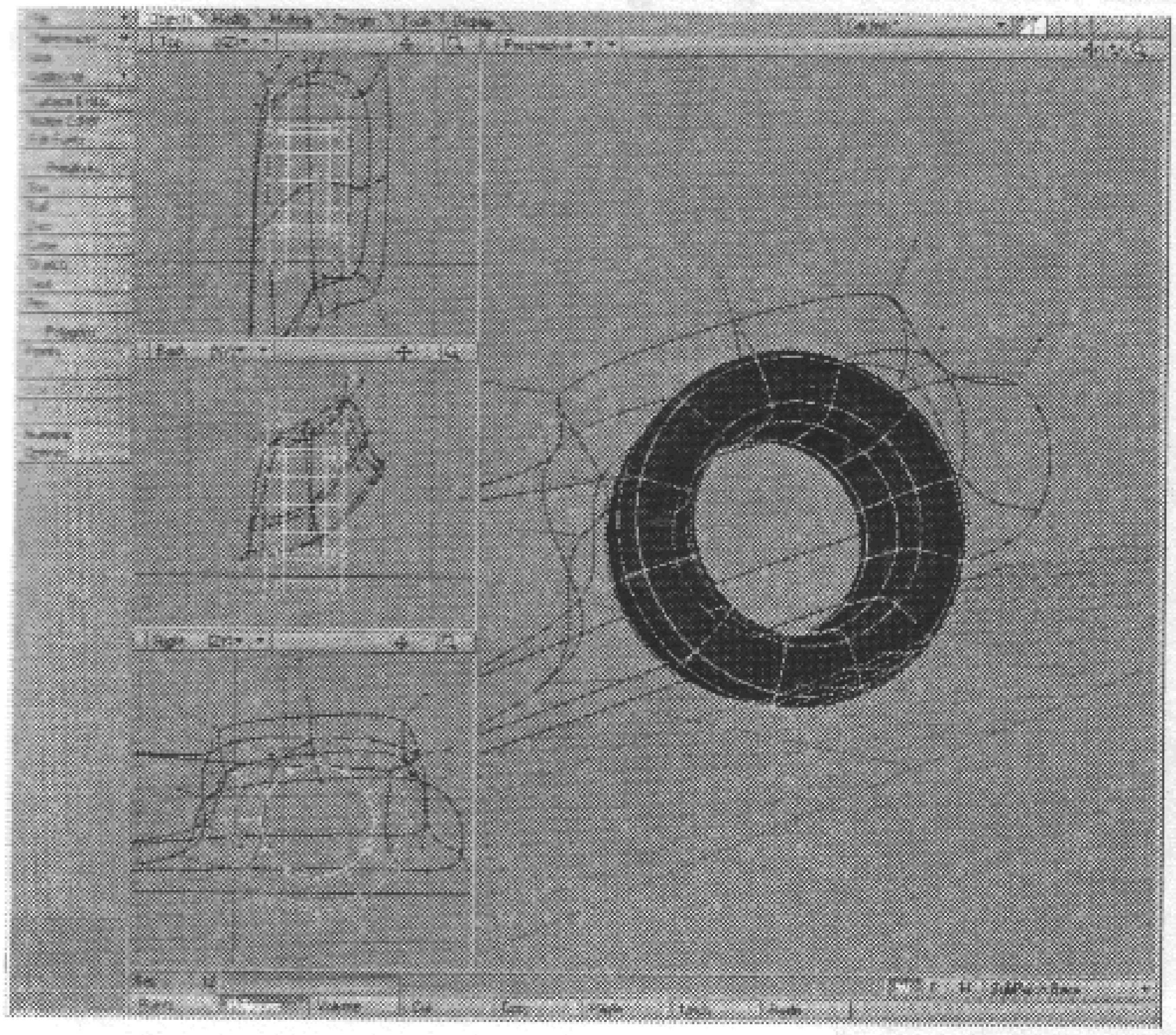
33 Then **Knife** at X 80cm and X 1.04m.

34 Drop the tool and select the **Lathe** tool. Let's do this operation numerically. In the Numeric panel, first select **Activate** from the **Actions** pop-up menu. Set the fields as follows: Sides 12, Axis X, Start Angle 0, End Angle 360, Center XYZ 62cm, -30cm, -1.66m, Offset 0. Drop the tool to apply the settings. (Note: We will be using these coordinates for **Lathe** regularly from here on out.)

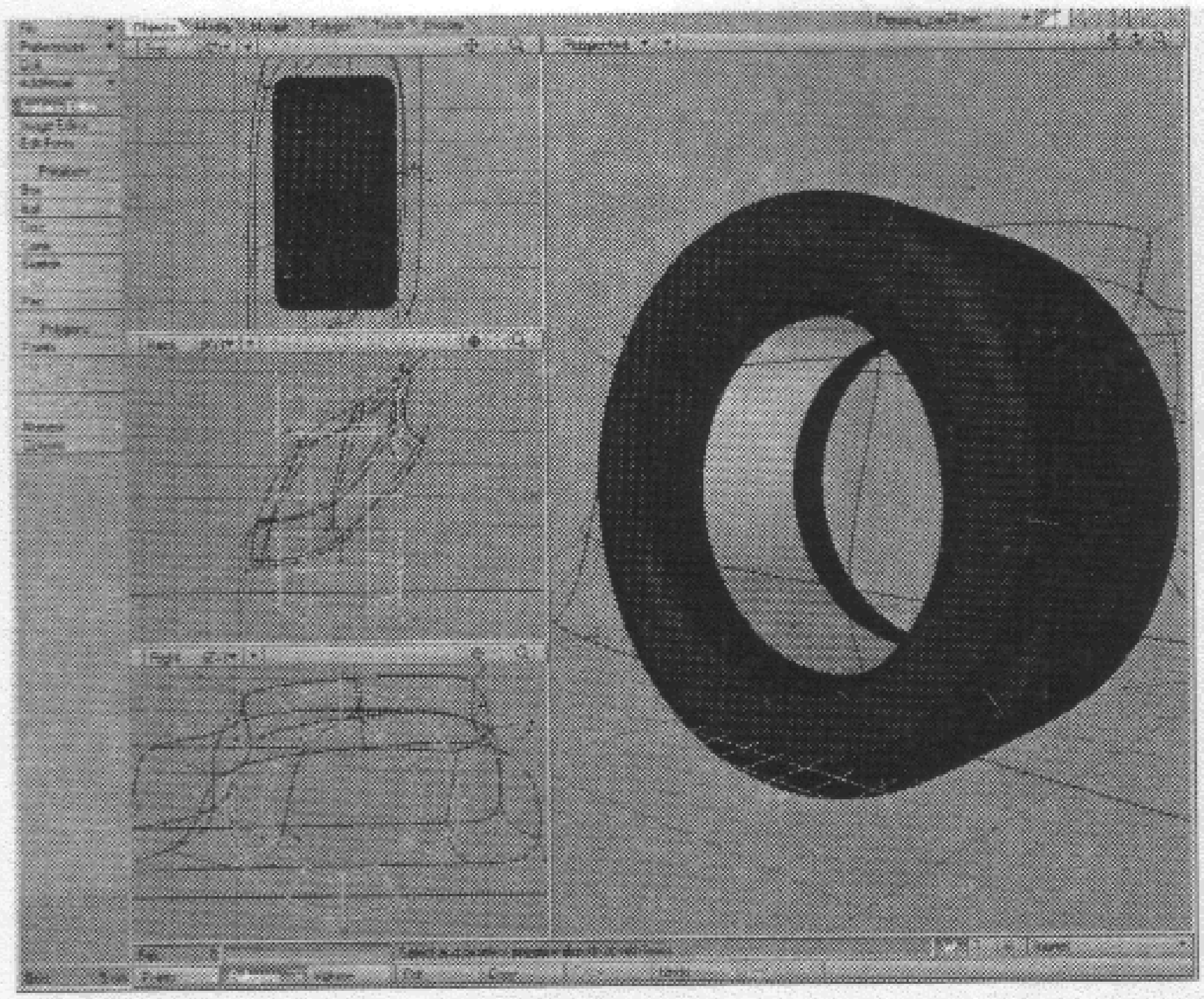
Press the TAB key to activate **SubPatch**. If the polygons look inside out, choose **Polygons** > **Flip** to flip them around. Name the surface TIRE.



35 Next place your mousepointer over the Right viewport and press the 0 key on your numeric keypad. That viewport will become full screen. Then choose **Display** > **Fit Sel** to fit the polygons into the display. Select just the 12 polygons along the inside. The easiest way is to drag vertically over the center of the tire in the Back viewport and then unselect the unwanted outer ones in the Right viewport. When finished, press keypad 0 and then **Fit Sel** again. Name the surface RIMS.



Open the Surface Editor and select the Tire surface. Use the following attribute settings: **Color**: red 020, green 000, blue 070, **Diffuse** 90%, **Specularity** 10% and **Smoothing** on. Select the Rims surface and use: **Diffuse** 90%, **Specularity** 75%, **Glossiness** 25% and **Reflectivity** 80%, with **Smoothing** on. Leave all other attributes at their default.



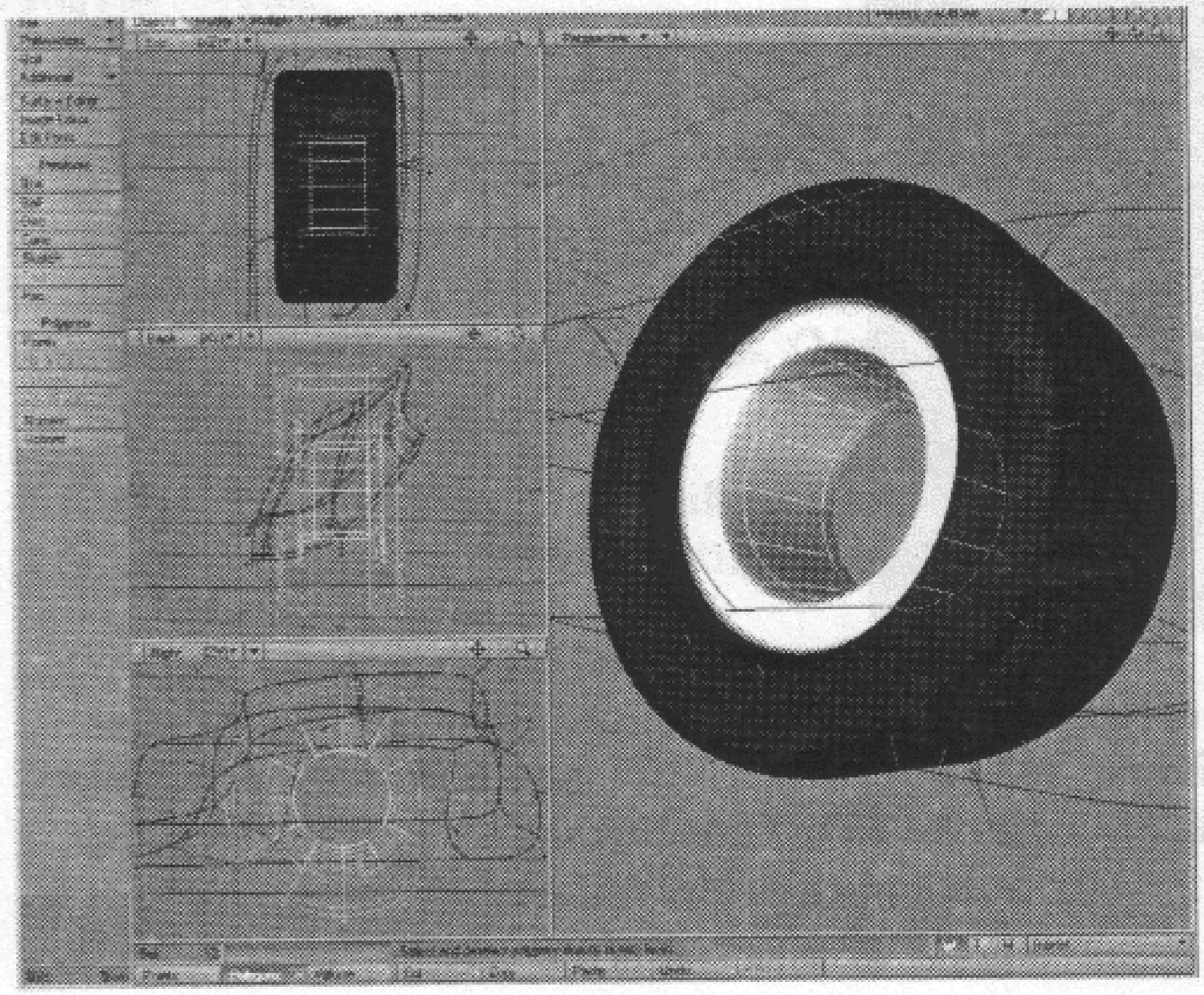
Your first tire

36 The Rims polygons should still be selected. **Smooth Shift** them with an **Offset** of 2cm and **Max Smoothing Angle** of 89.5.

37 Select the **Stretch** tool and in the Back viewport position your mousepointer at X 82m Y 30cm. **Stretch** the **Horizontal** to 110% and 100% **Vertical**.

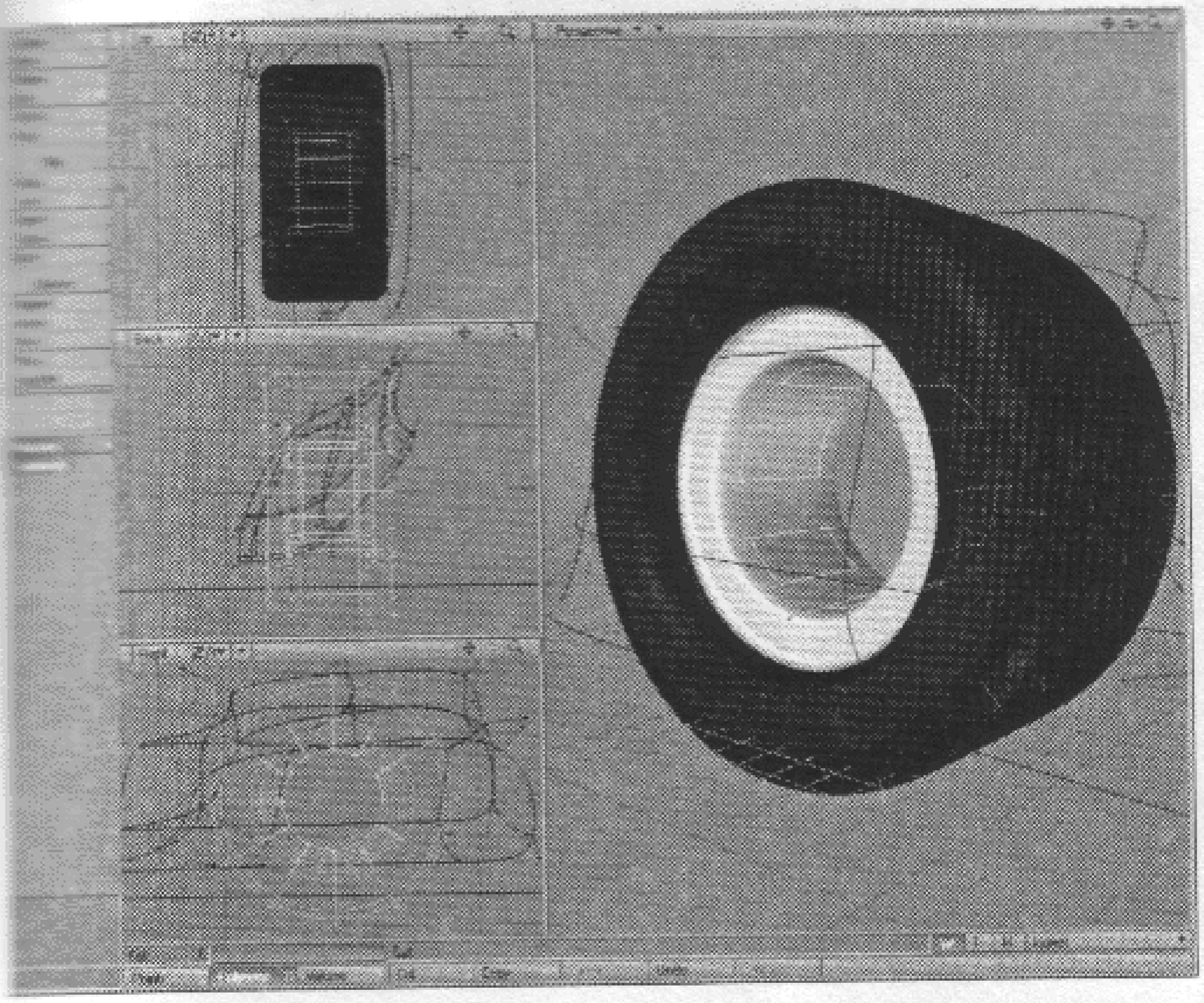
38 **Smooth Shift** with an **Offset** of 2cm and **Max Smoothing Angle** of 89.5. We want to do this two more times numerically, so drop and re-select **Smooth Shift**—this applies the operation and restarts.

Then select **Activate** from the **Actions** menu on the Numeric panel to use the same settings again. Drop and re-select **Smooth Shift** again. Select **Activate** again and drop the tool for a final time.

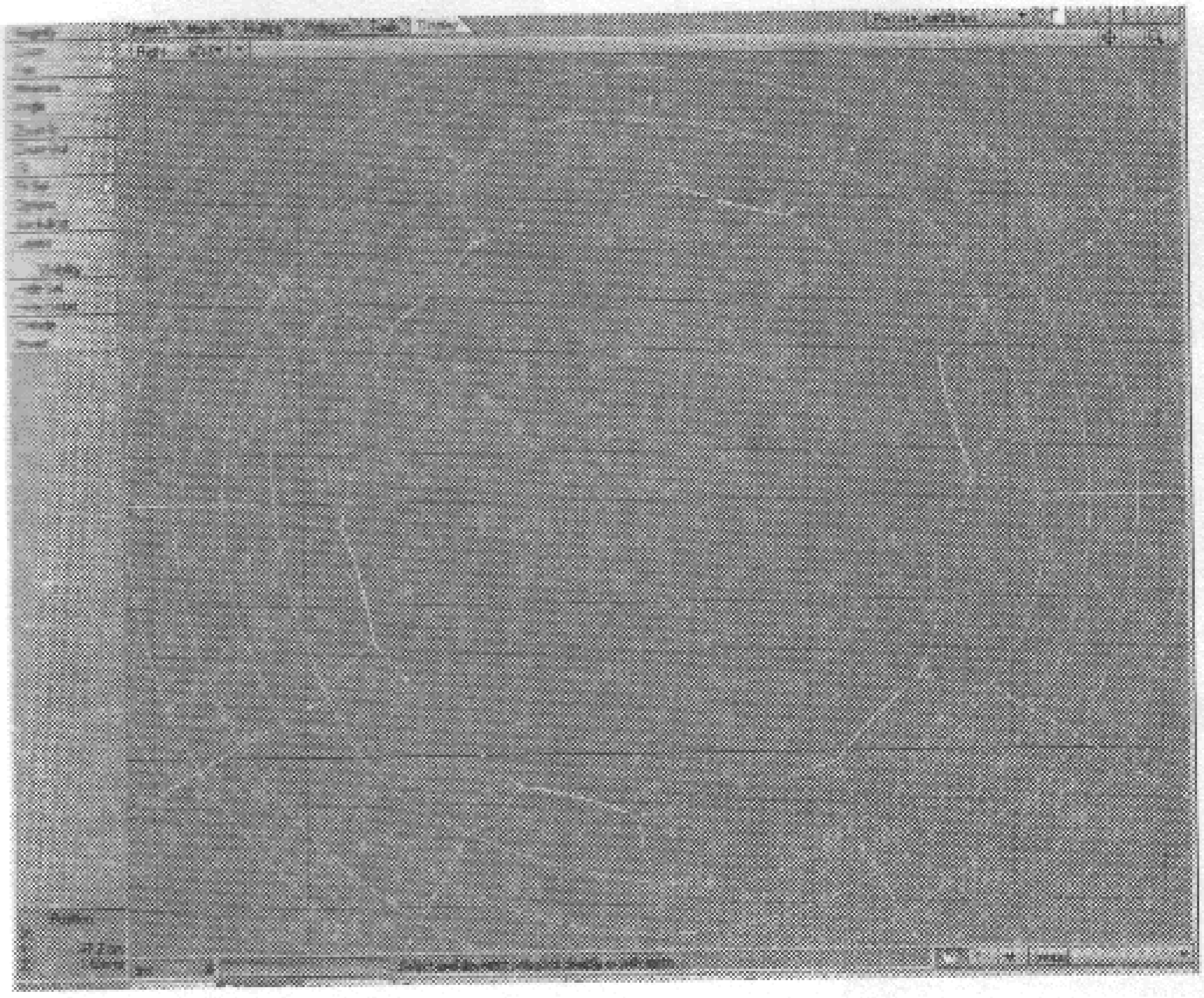


39 Select the **Stretch** tool and in the Back viewport place your mousepointer at X 82cm Y [illegible]cm. **Stretch** the selection to 90% **Horizontal** and 100% **Vertical**.

40 In the Right viewport, deselect every other Rims polygon.



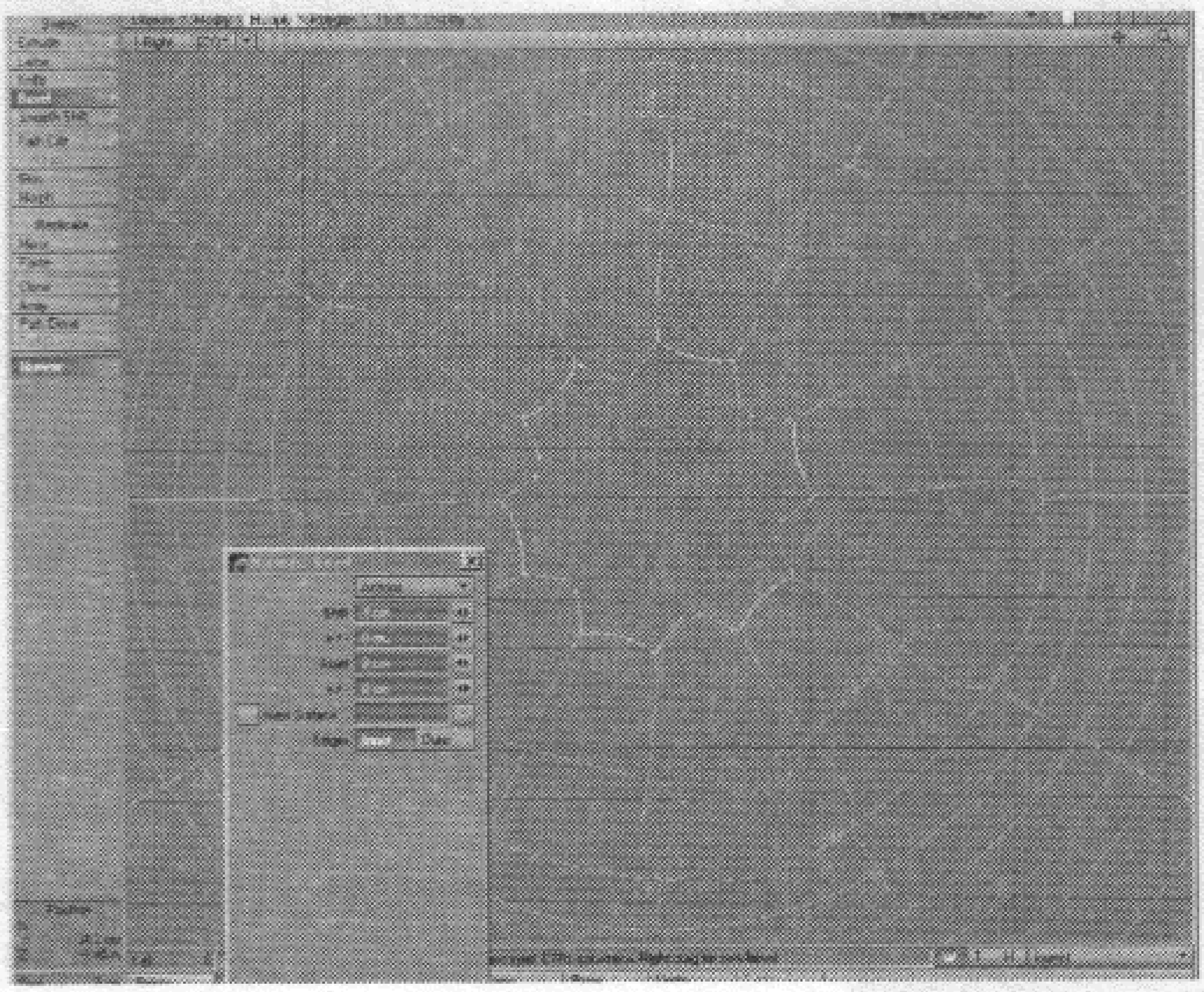
41 Choose **Multiply** > **Bevel**. Drag your RMB in the Right viewport so that **Shift** is 4cm and **Inset** is 0cm. Click your RMB to bevel some more.



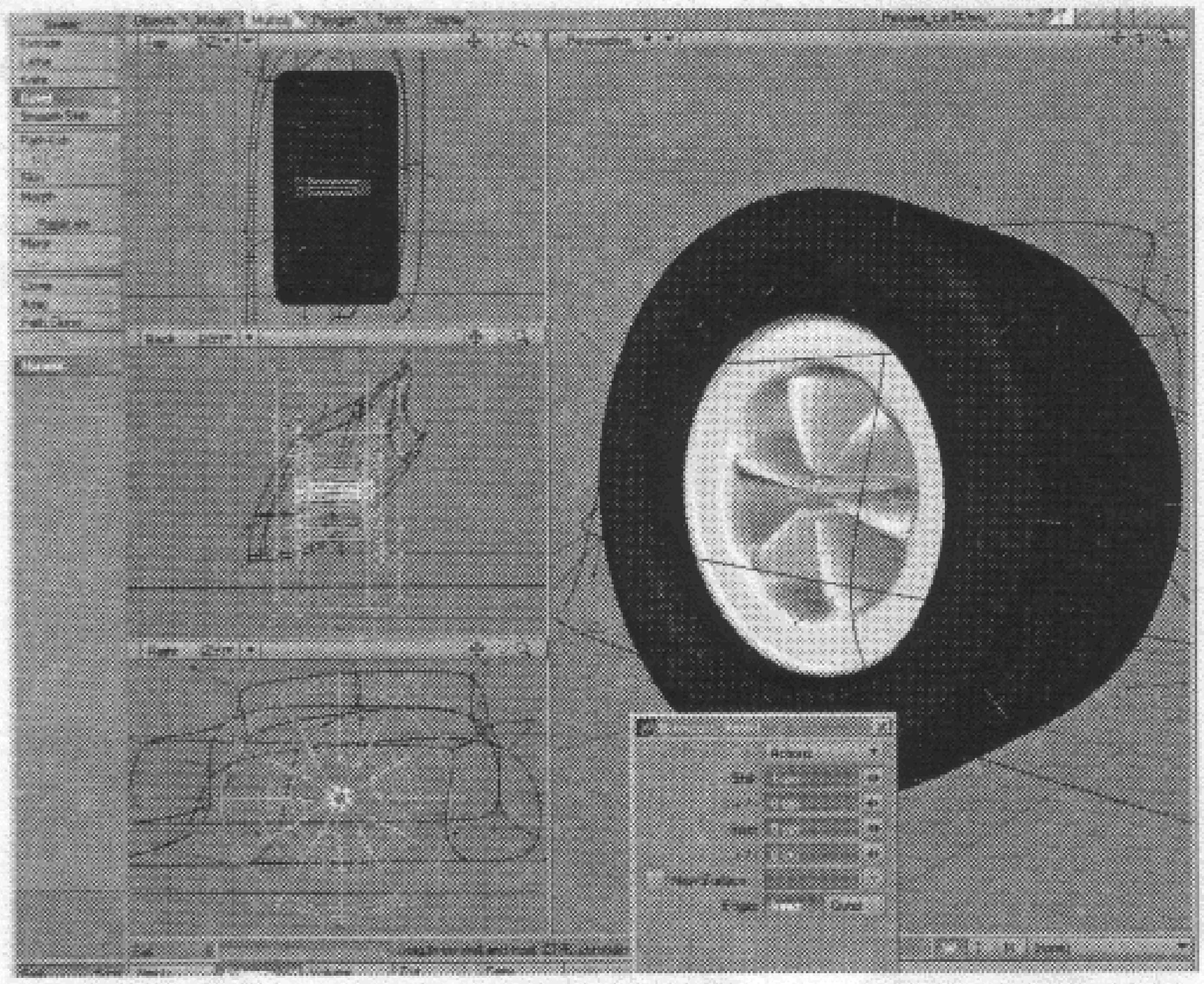
☞ *NOTE*

Smooth Shift and **Bevel** are very similar; however, **Smooth Shift** generally treats contiguous polygons as a group, where **Bevel** treats each polygon individually.

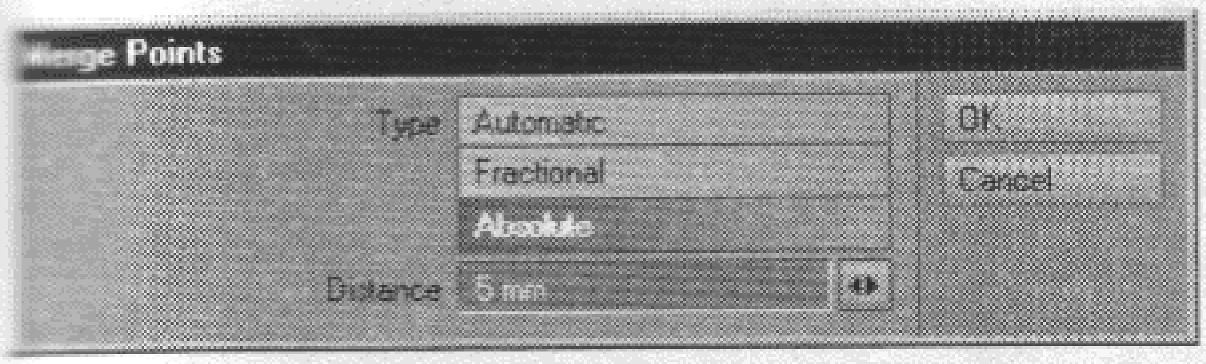
42 Still using the **Bevel** tool, drag your RMB in the Right viewport so that the points to either side of each polygon almost touch their neighbors. **Shift** should be about 7cm and **Inset** is 2cm. (Dragging down will change the **Inset**.)



Click your RMB to accept the values and continue beveling. Bevel some more with 0 Inset until the edge points actually overlap. **Shift** should be around 8mm. Drop the tool to accept the values.

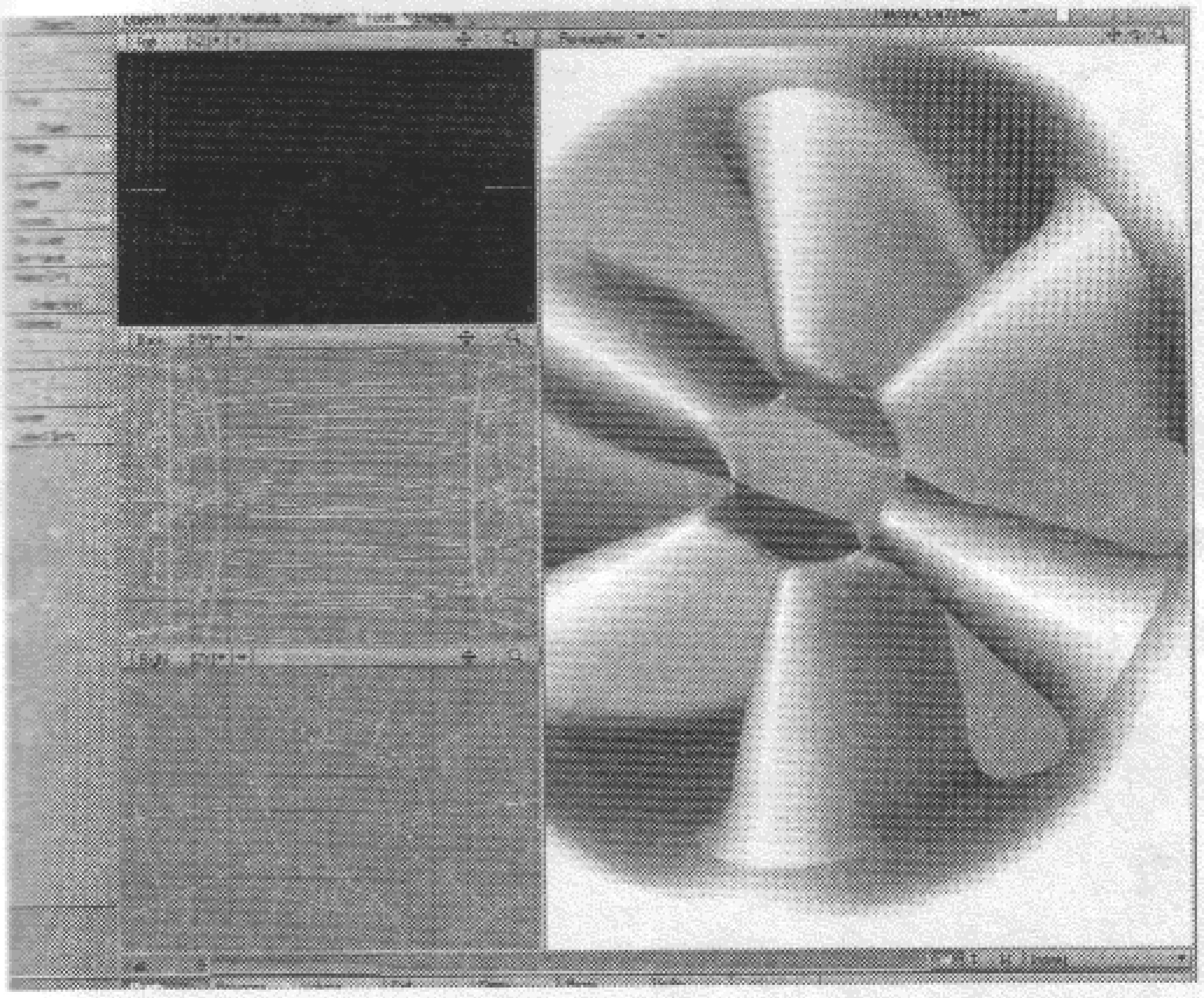


43 Switch to **Point** Selection mode. Choose **Tools** > **Merge** and set the Merge Points settings to **Absolute** and **Distance** 5mm. Click **OK** to join the spokes to each other. Twelve points should be eliminated.


Merge Points

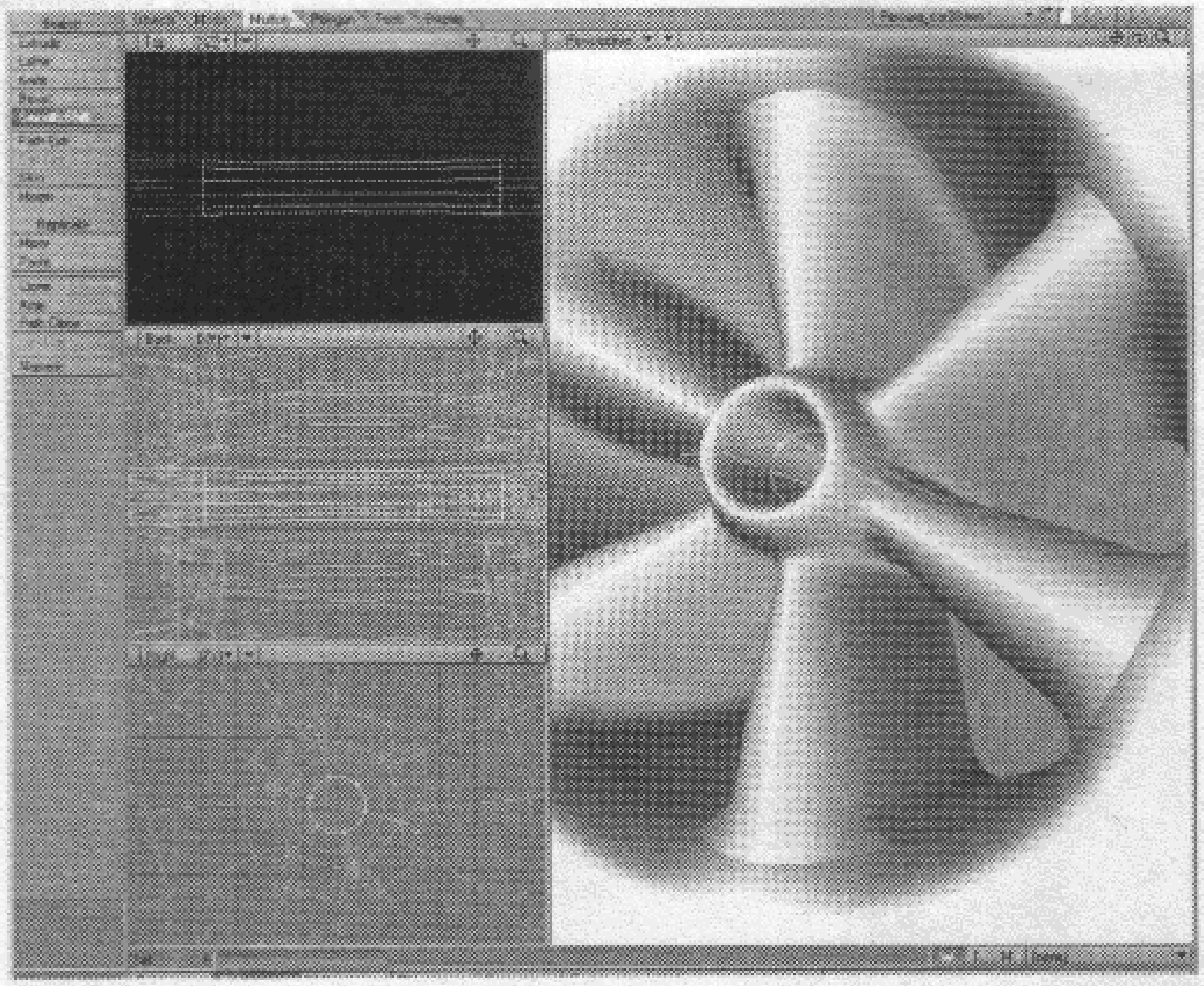
Type: Automatic / Fractional / Absolute

Distance: 5 mm

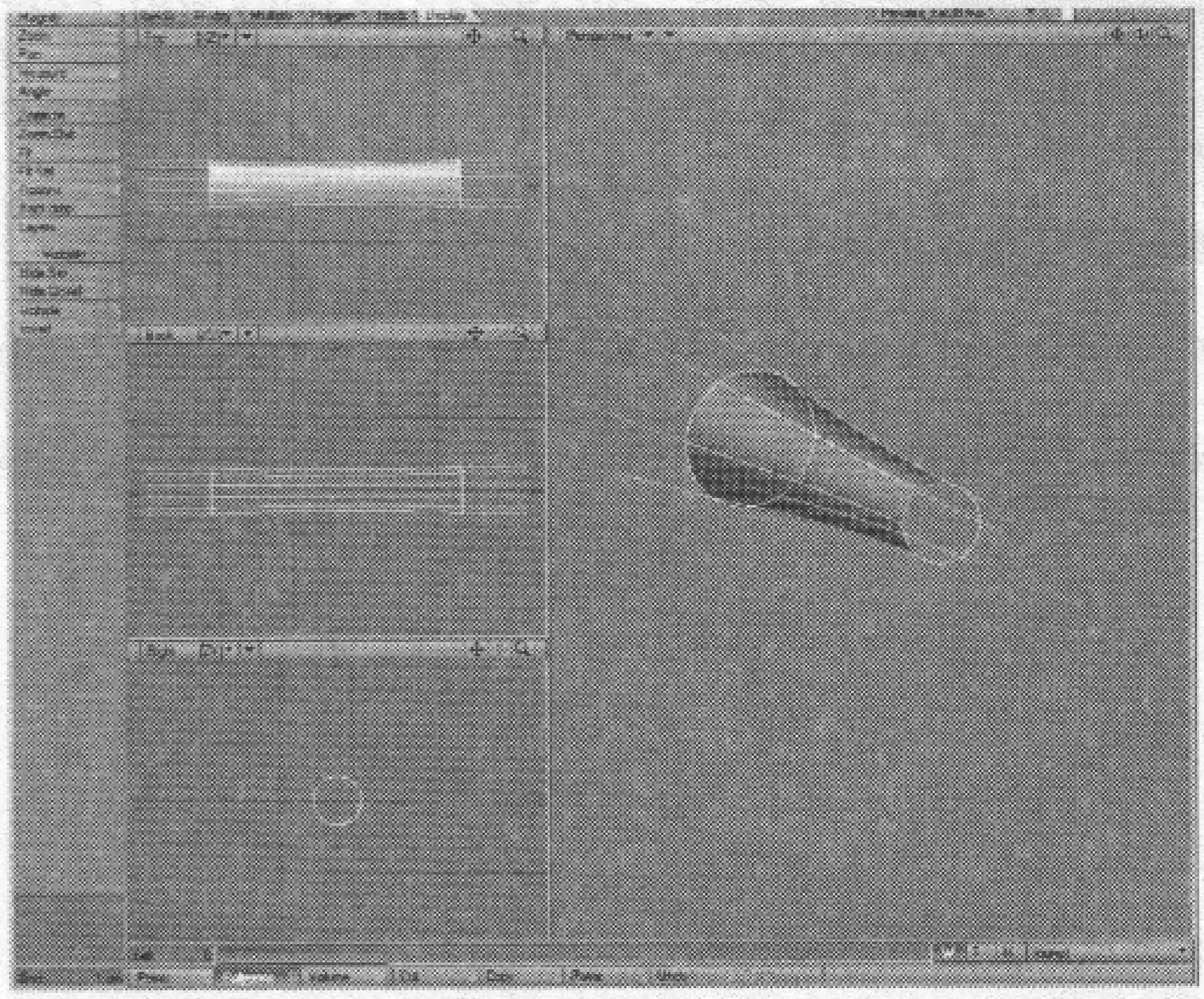


44 Switch to the **Polygon** Selection mode. The six inside polygons should still be selected. **Smooth Shift** to an **Offset** of 2cm and **Max Smoothing Angle** of 180. Using techniques described previously, repeat one more time with **Offset** of 1.8cm and **Max Smoothing Angle**

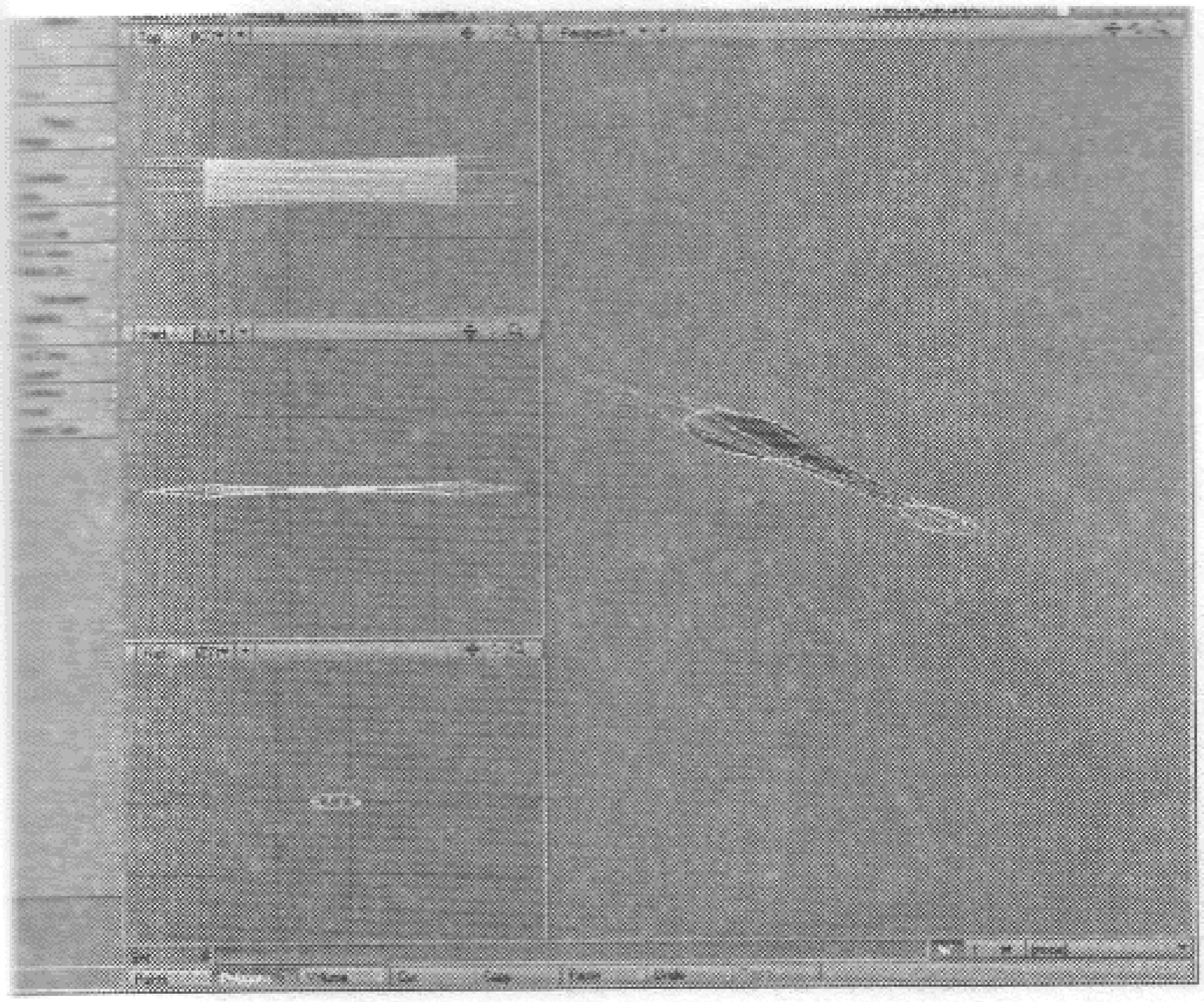
of 90. (Note: Reducing the **Max Smoothing Angle** keeps the perpendicular polygons from affecting the direction of the **Offset**.)



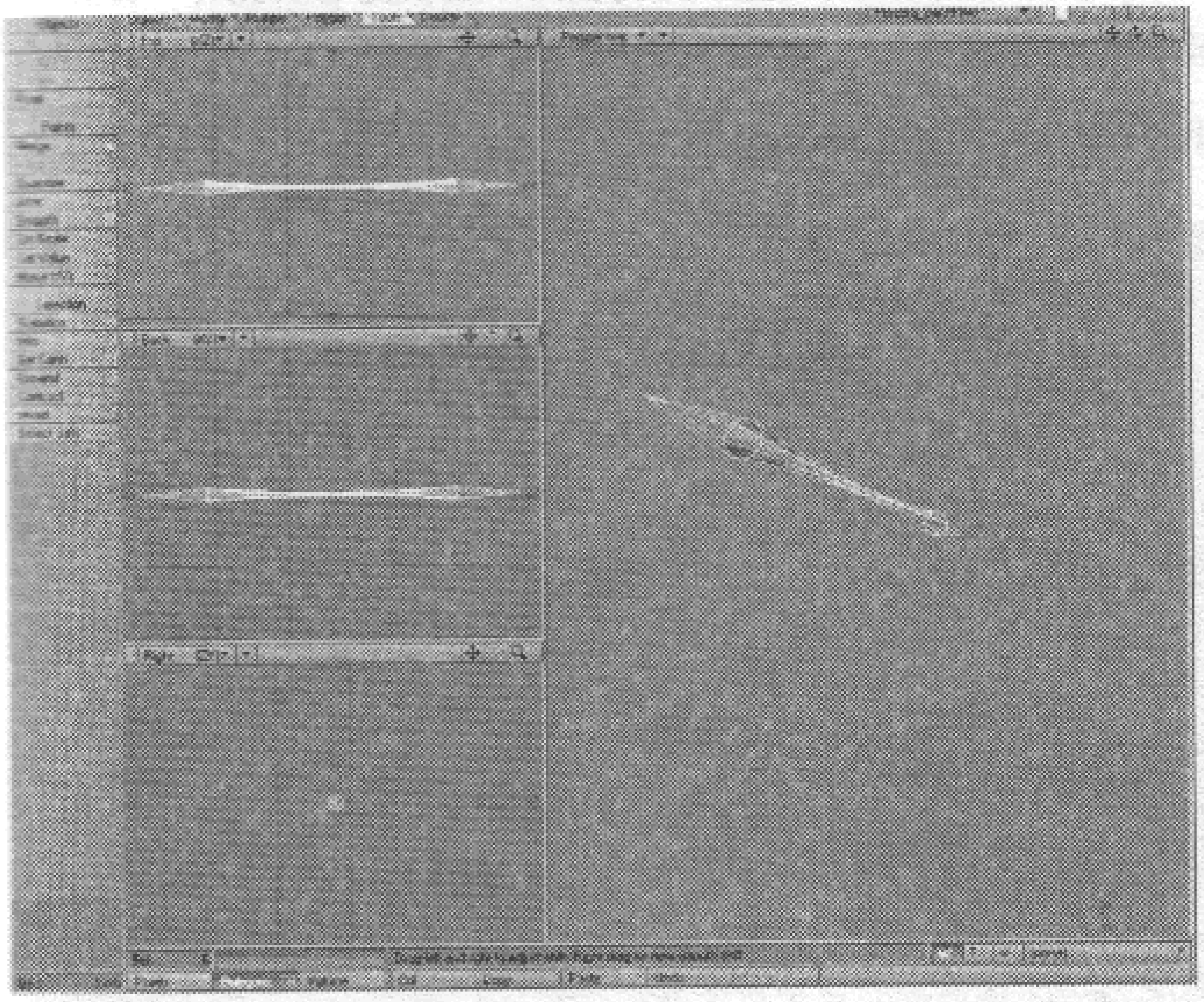
45 Choose **Display** > **Hide Unsel**, so that only the inside polygons are visible.



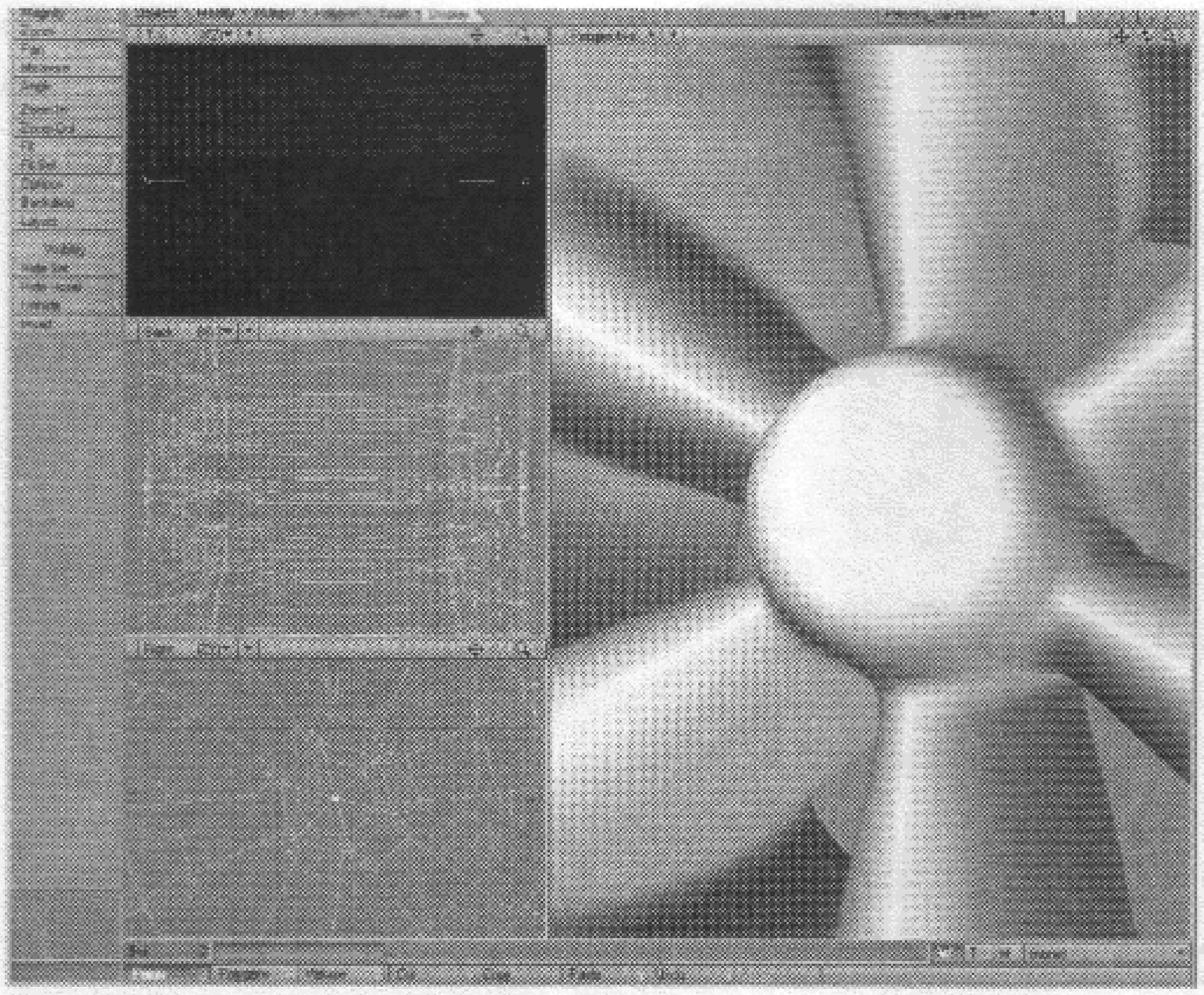
Let's move all of these points to the same position. Choose **Tools** > **Set Value**. Apply with **Axis Y** set to 30cm.



Use Set Value again with **Axis Z** set to -1.66m. Note that since we didn't change the X axis value the points stay apart along the X axis.

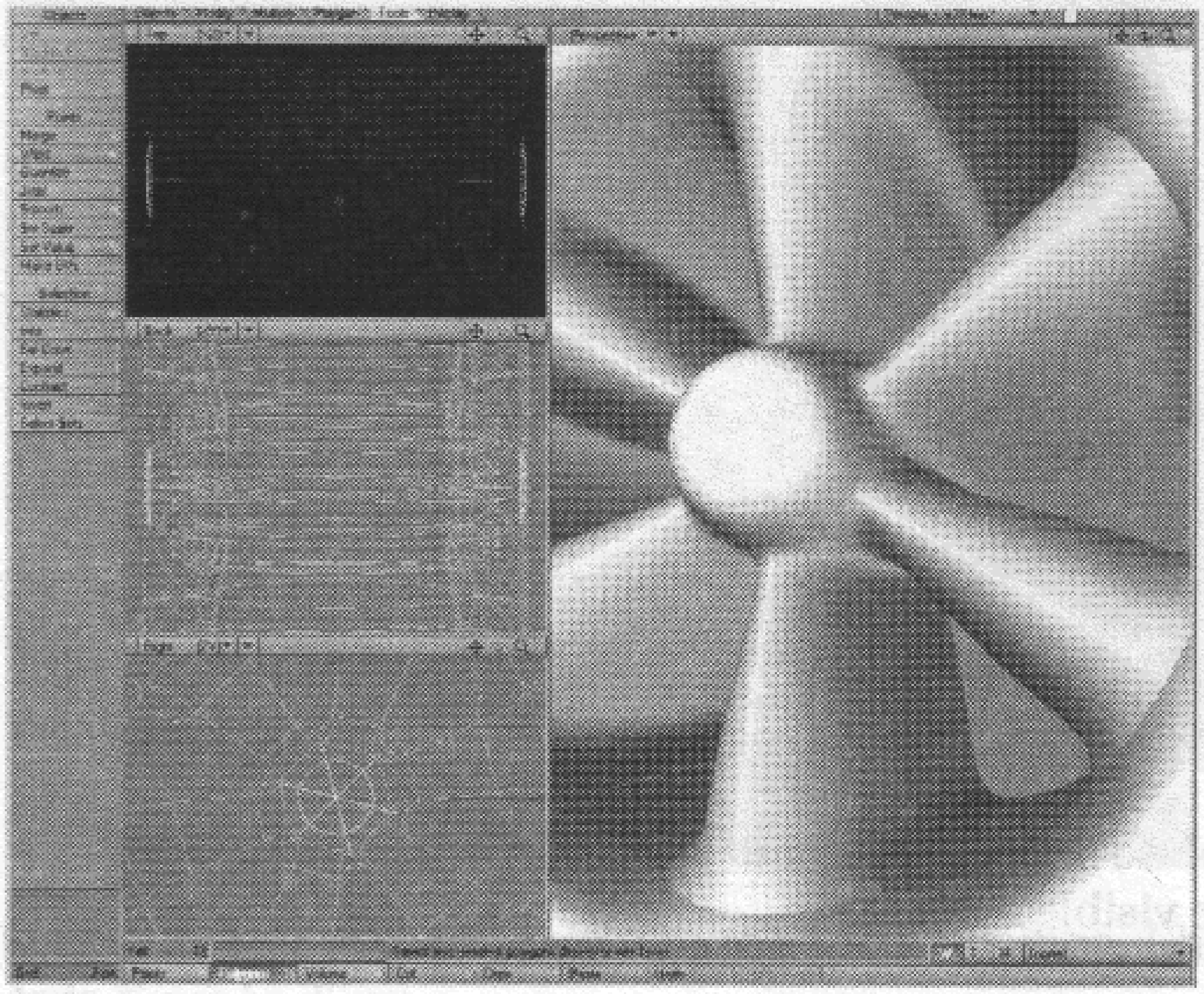


46 Now use **Merge Points** set to **Absolute** 5mm. Ten points should be eliminated. Choose **Display** > **Unhide** make the whole tire visible again.

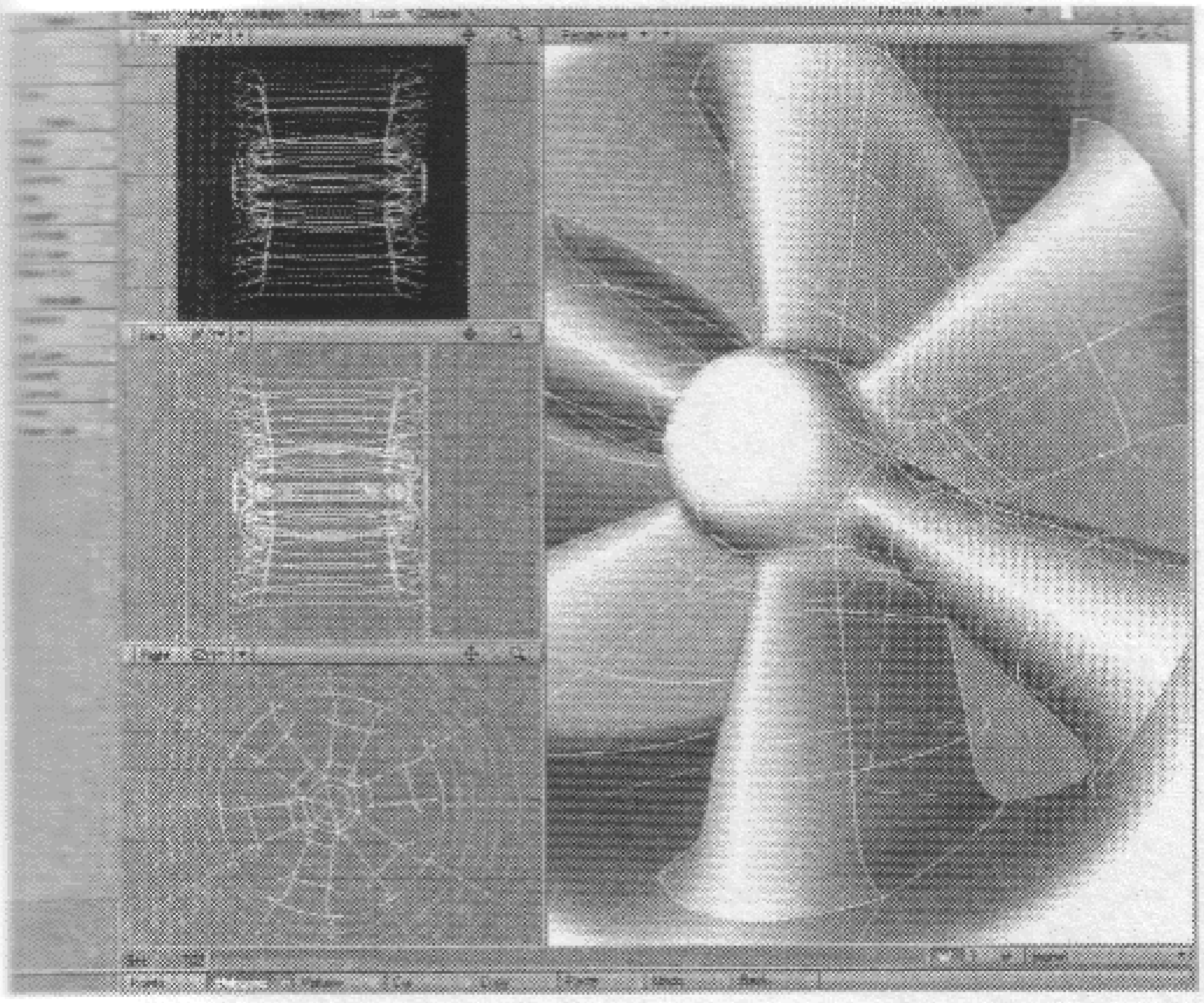


After you merged the points, all the polygons you had selected are now condensed into two two-point polygons. Although you can't see this, they should still be selected. You can tell because the Selection count display—right about the **Point** Selection mode button—will say 6. Choose **Polygon** > **Remove** to get rid of them.

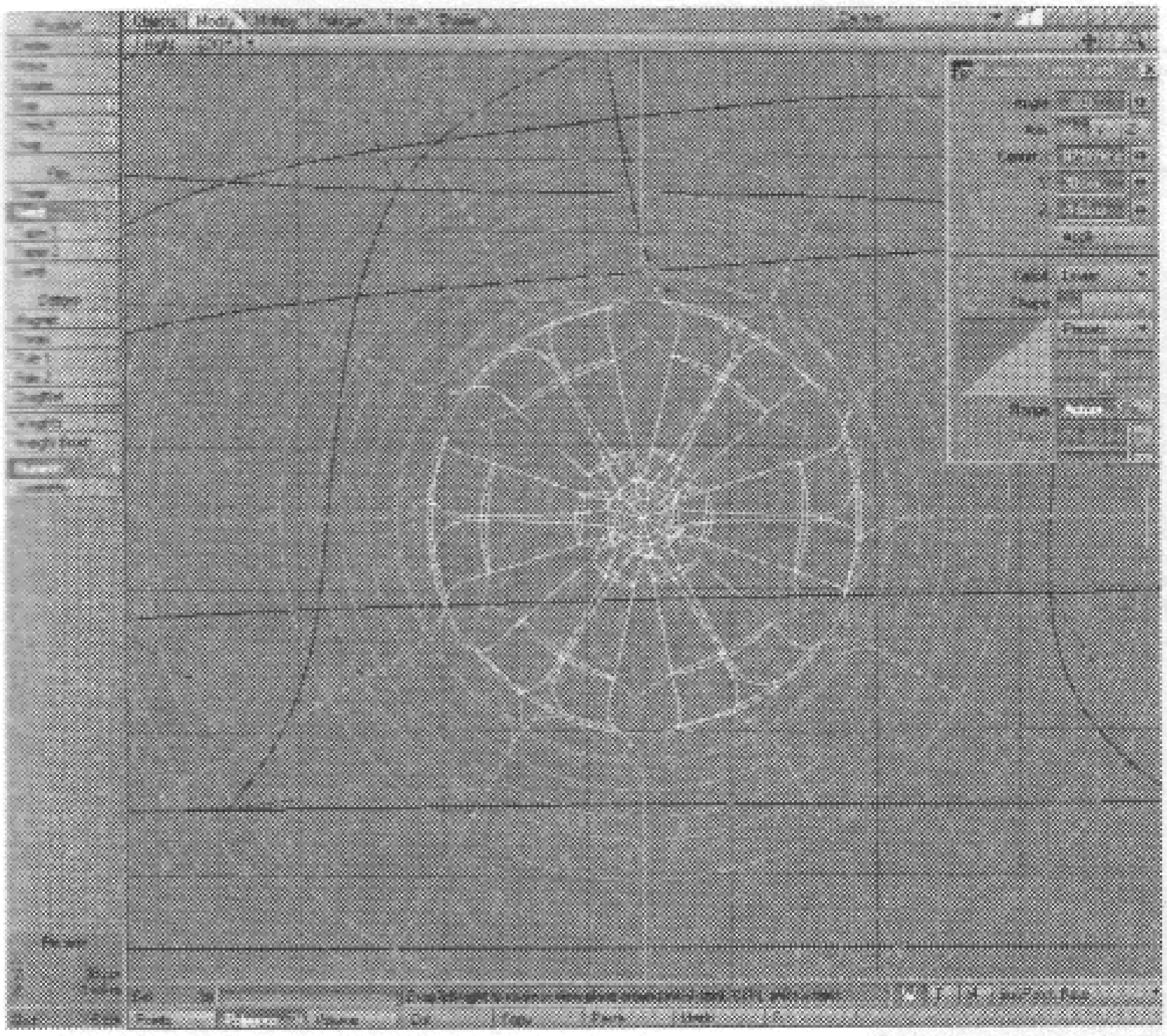
47 Select the inner-most polygons—should be 12 polygons.



Click **Tools** > **Expand** to increase the selection until you get to the top of the spokes. This should take four clicks.

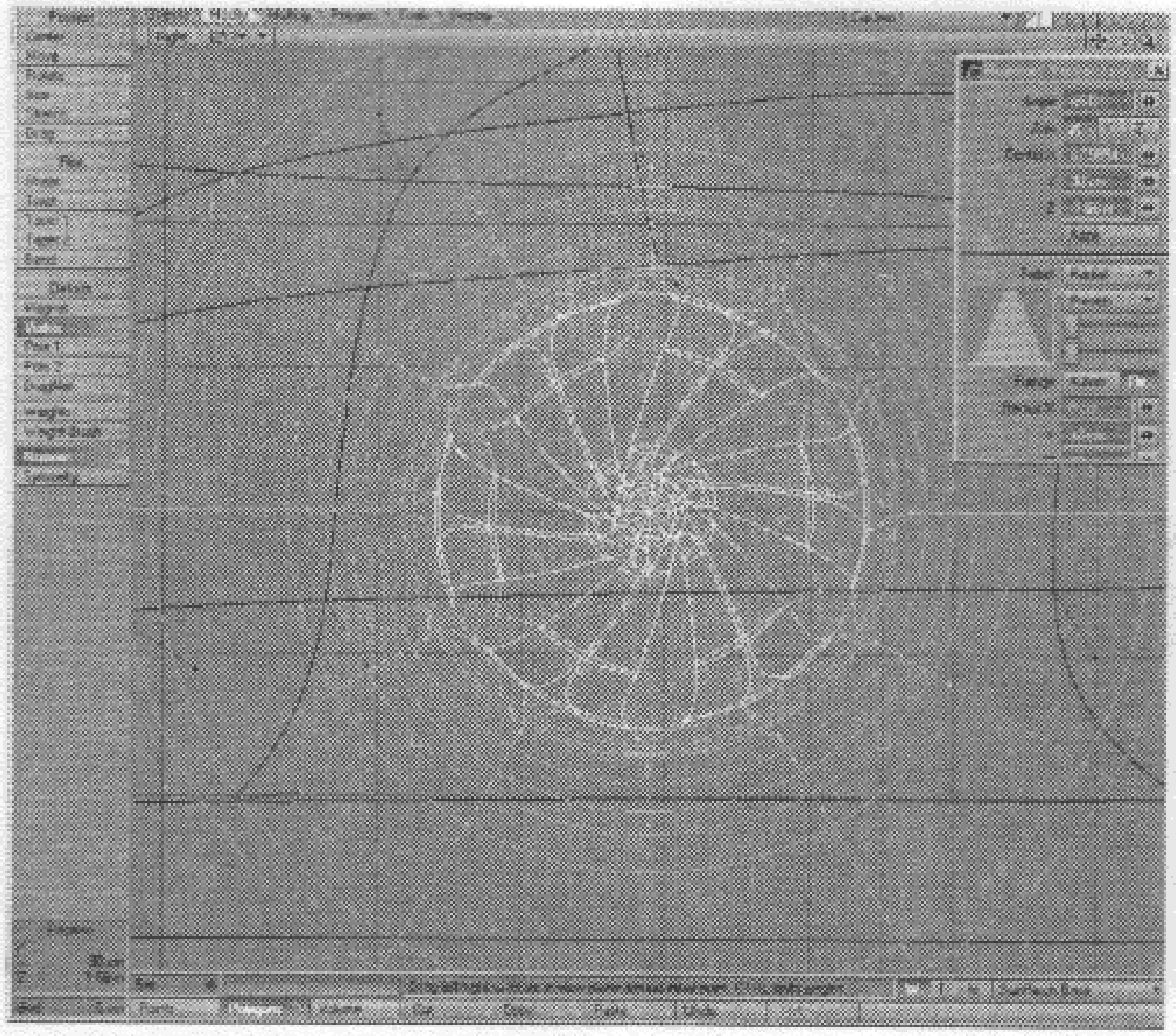


48 Select **Modify** > **Twist** and in the Right viewport twist the selection 30 degrees, with your cursor centered at Y 30cm and Z -1.66m.



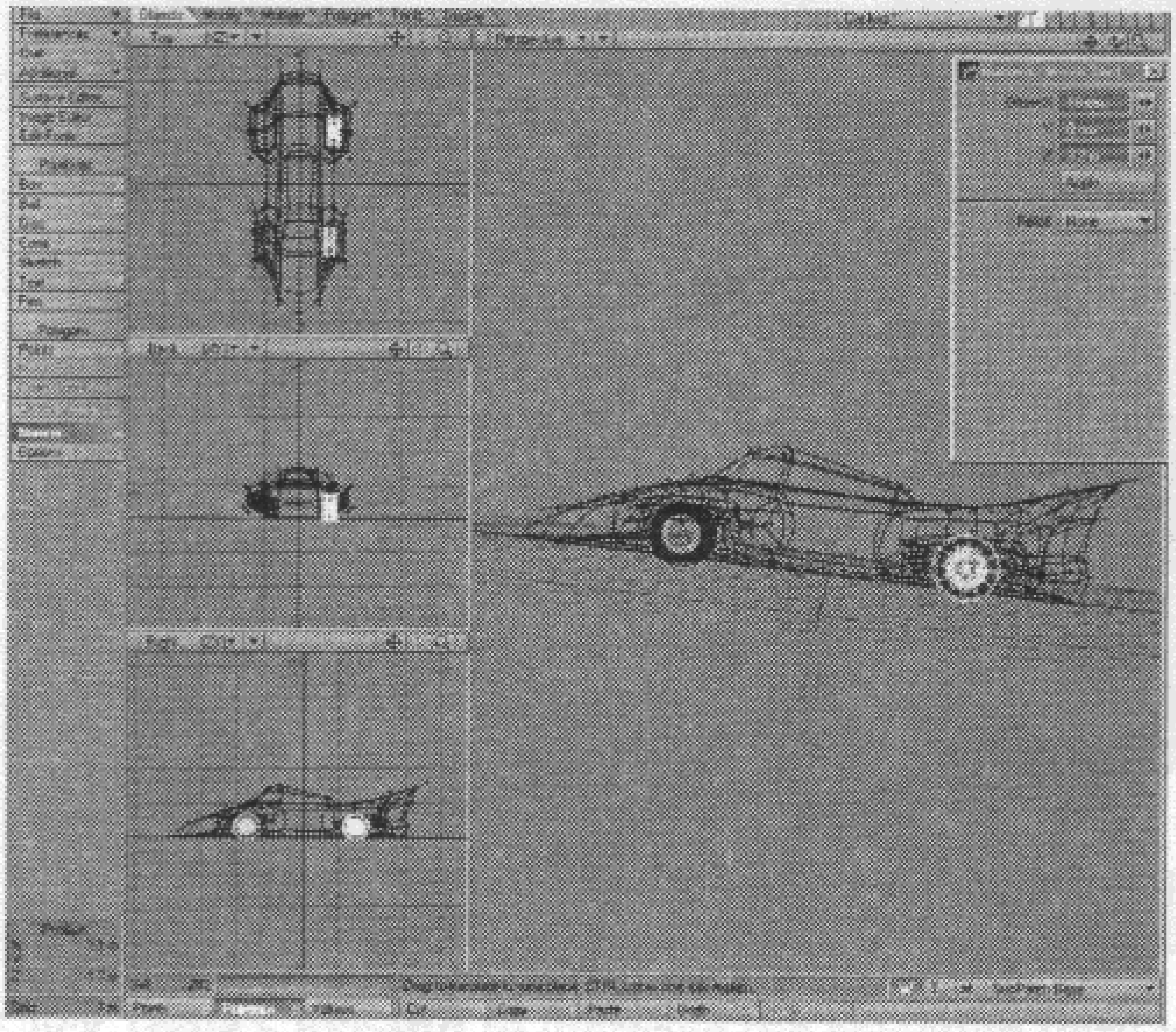
Making cool rims

49 Select **Modify** > **Vortex** and in the Right viewport vortex the selection 45 degrees with your cursor centered at Y 30cm and Z -1.66m.

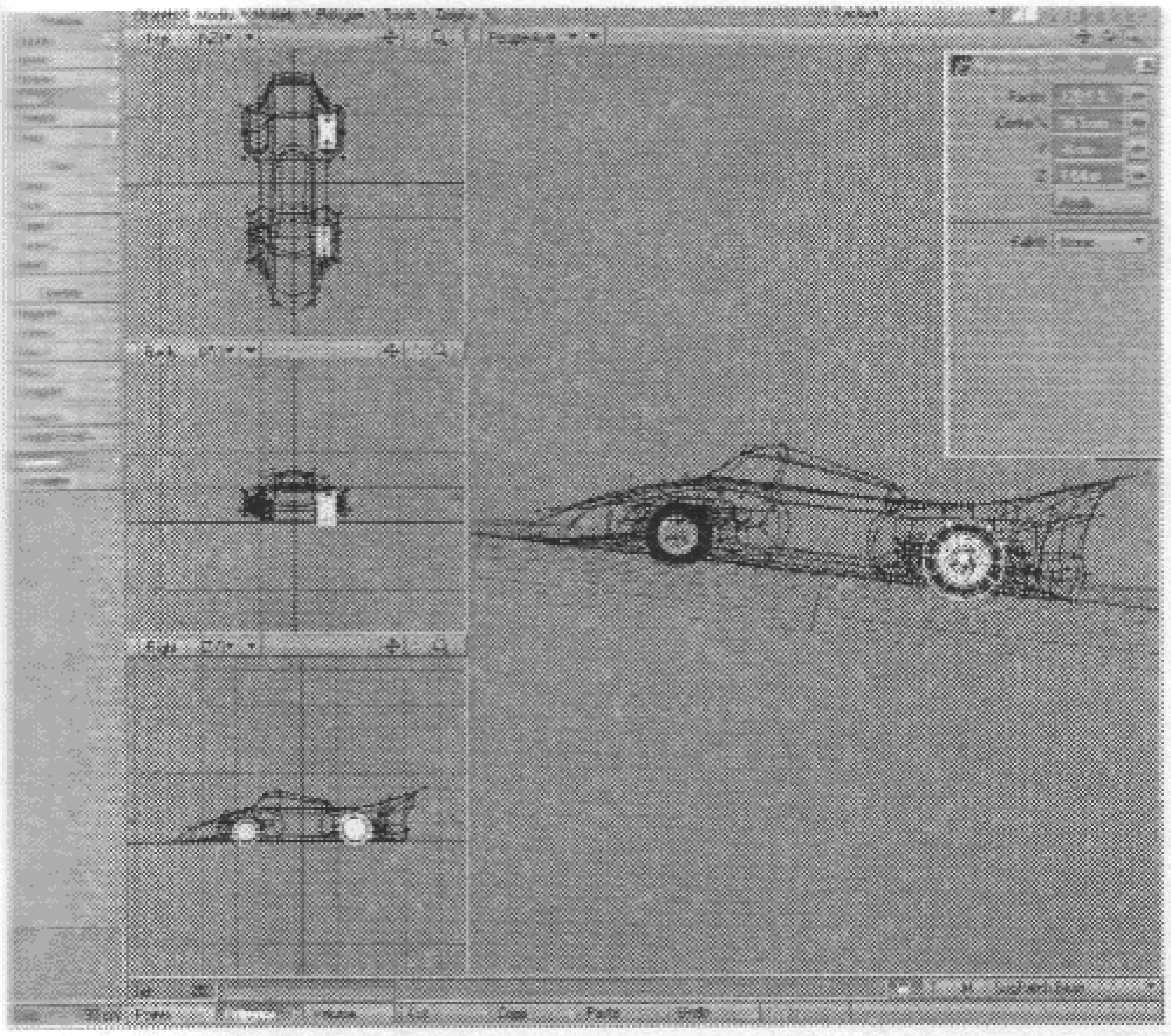


50 Go to Layer 1 and choose **Display** > **Unhide** to make all of the car's polygons visible, then return it to the background.

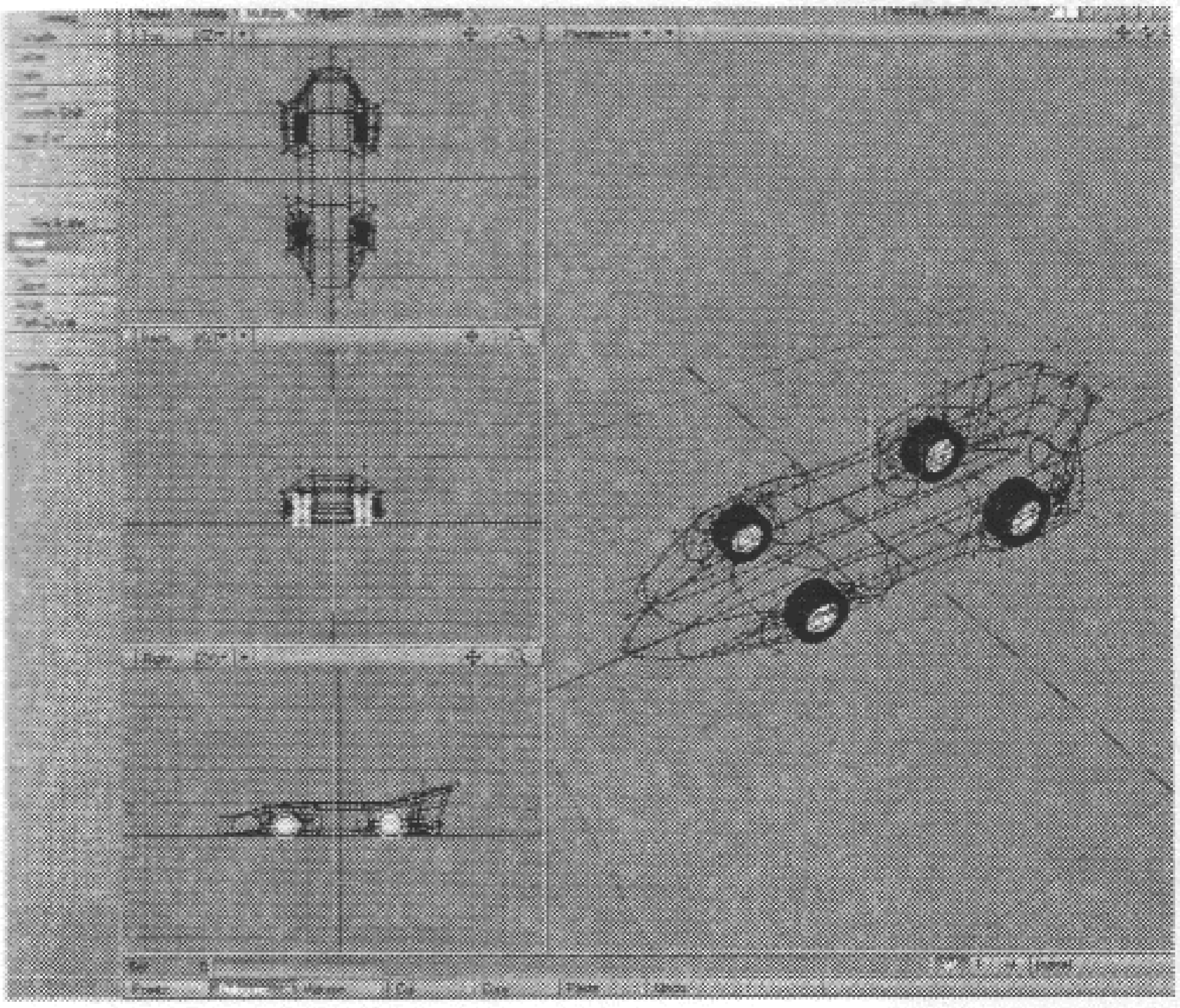
51 Highlight the entire tire object. Click the **Copy** button and then the **Paste** button—you won't notice much difference since the pasted geometry is right on top. **Move** the object back into position for the rear tire at approximately Z 3.2m.



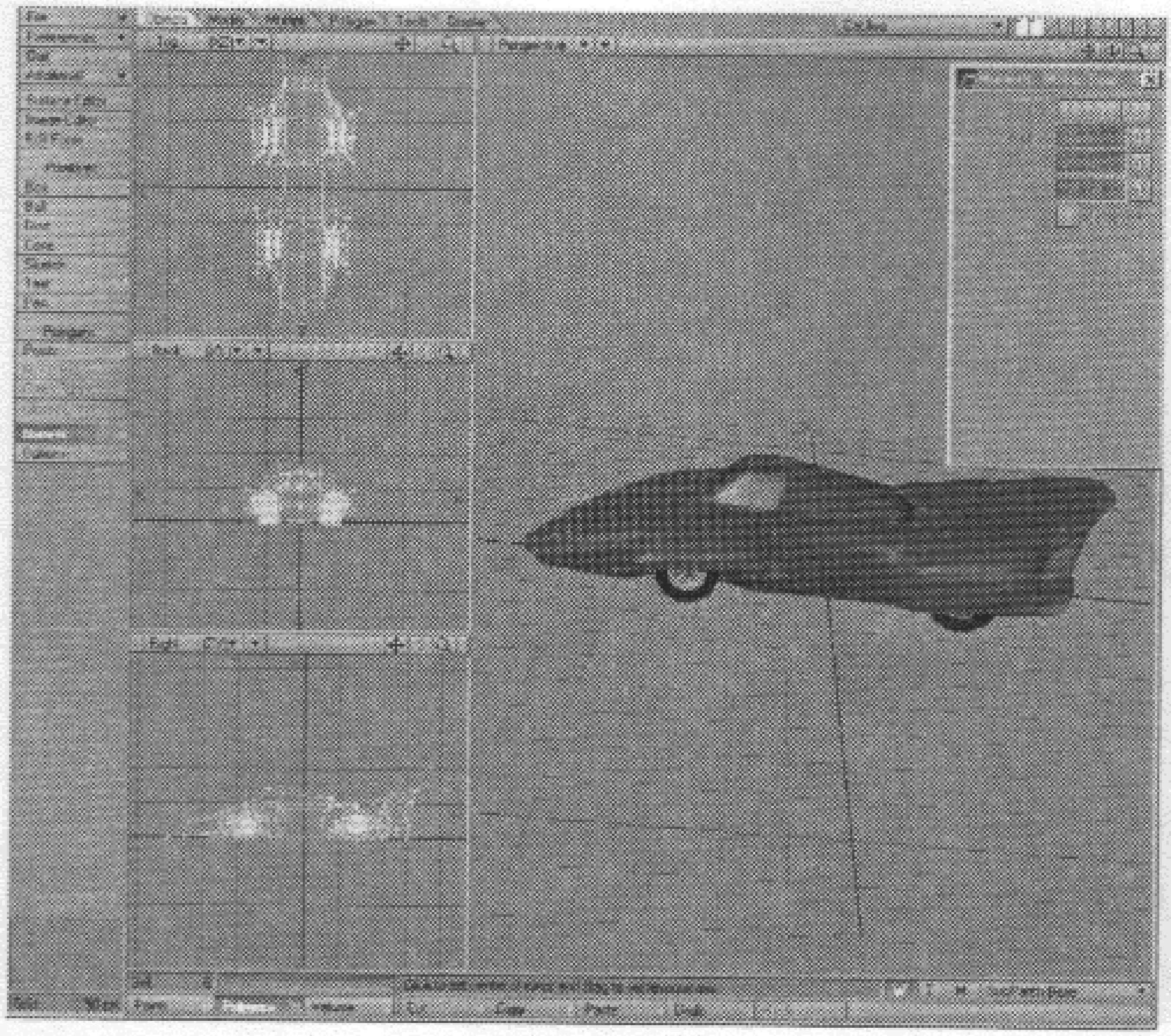
52 Select **Modify** > **Size** and in the Right viewport place your mousepointer at the bottom of the still selected rear tire—this should be about Y -8cm Z 1.54m. Size up the tire by 120%. Drop the tool and deselect the tire.



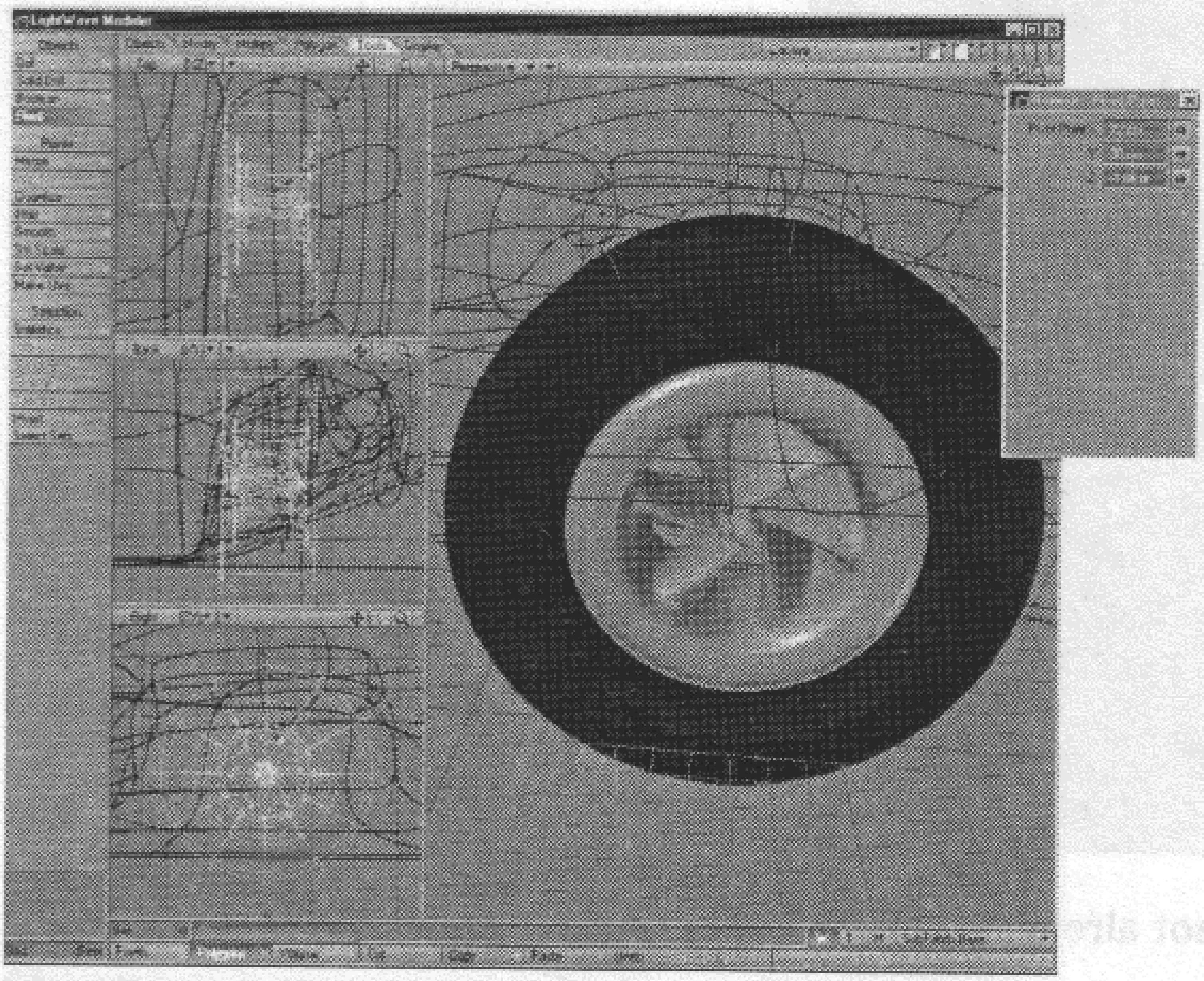
53 Choose Multiply > **Mirror** and in the Back viewport drag out a line along the Y axis. You should end up with both tires mirrored to the other side.



54 Save your object if you have not already done so.

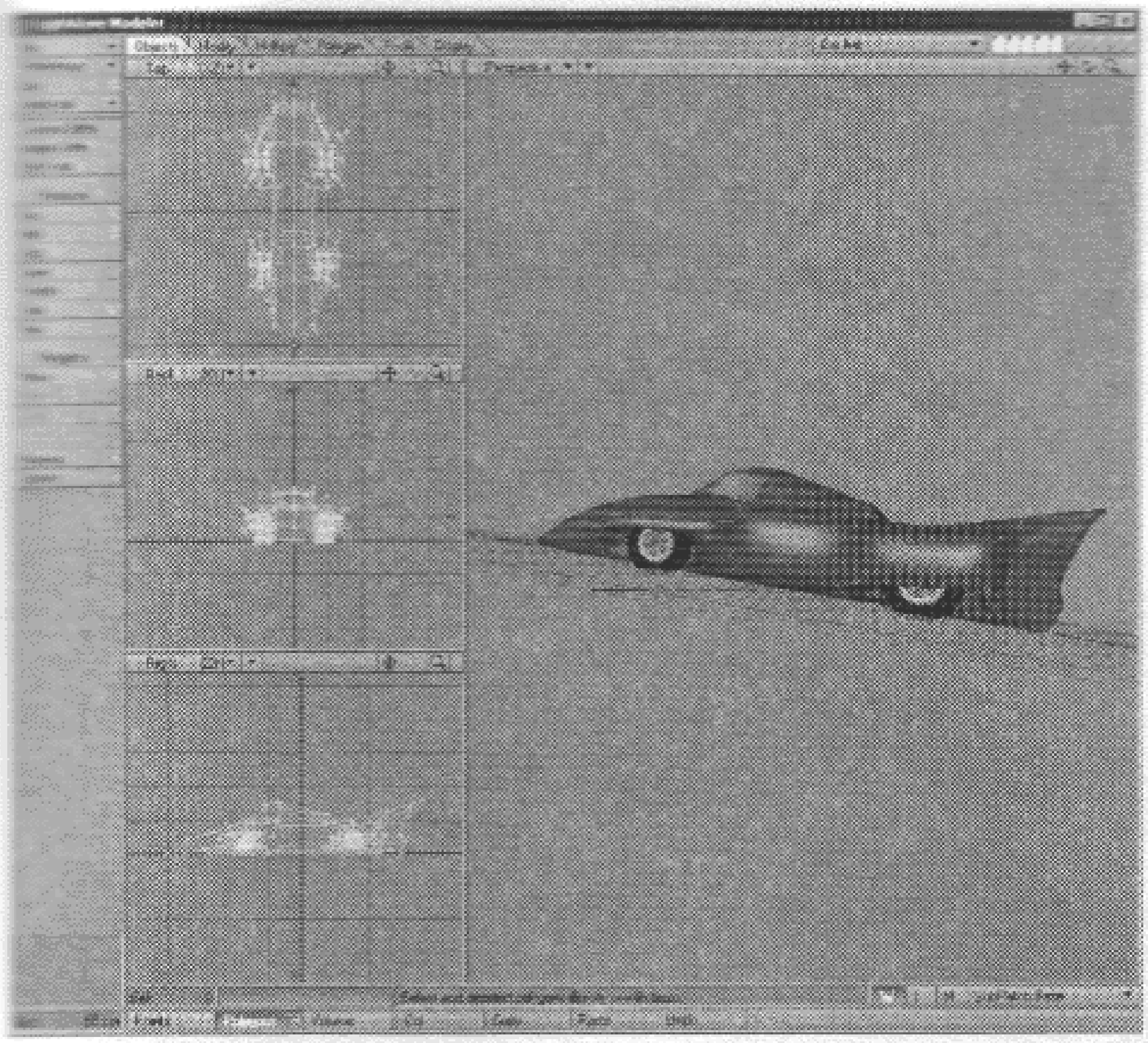


55 Make sure the Top viewport is in Wireframe mode. Then use the lasso (RMB drag) to select polygons to **Cut** and **Paste** three of the tires into empty layers. (If you don't use the Wireframe mode, you can't select the hidden polygons.)



Separating the tires

56 Choose **Tools** > **Pivot** and move the pivot points of each tire to the inside edge, aligned with the center. This will allow you to easily turn and spin each tire independently, if you choose to do so.



The Pecora Love machine ready to pick up chicks

57 Save your Object, then go to Layout and load it up.

58 Parent your tires to the body of the car. Open the Scene Editor. Select Layers 2, 3, 4, and 5 by holding down the CTRL key while clicking each layer. Drag the selection up to Layer 1. A yellow line will appear, when that line slides to the right release the mouse and the layers will be parented.

THE BASICS OF LIGHTING

In this chapter we will cover the basics of lighting a scene in Lightwave. We will not only cover how lights work but also some of the basic principals of lighting.

Lighting is one the most important elements in any animation. Lighting helps set the mood of any animation or image. Without proper lighting, an animation is at best flat and dull.

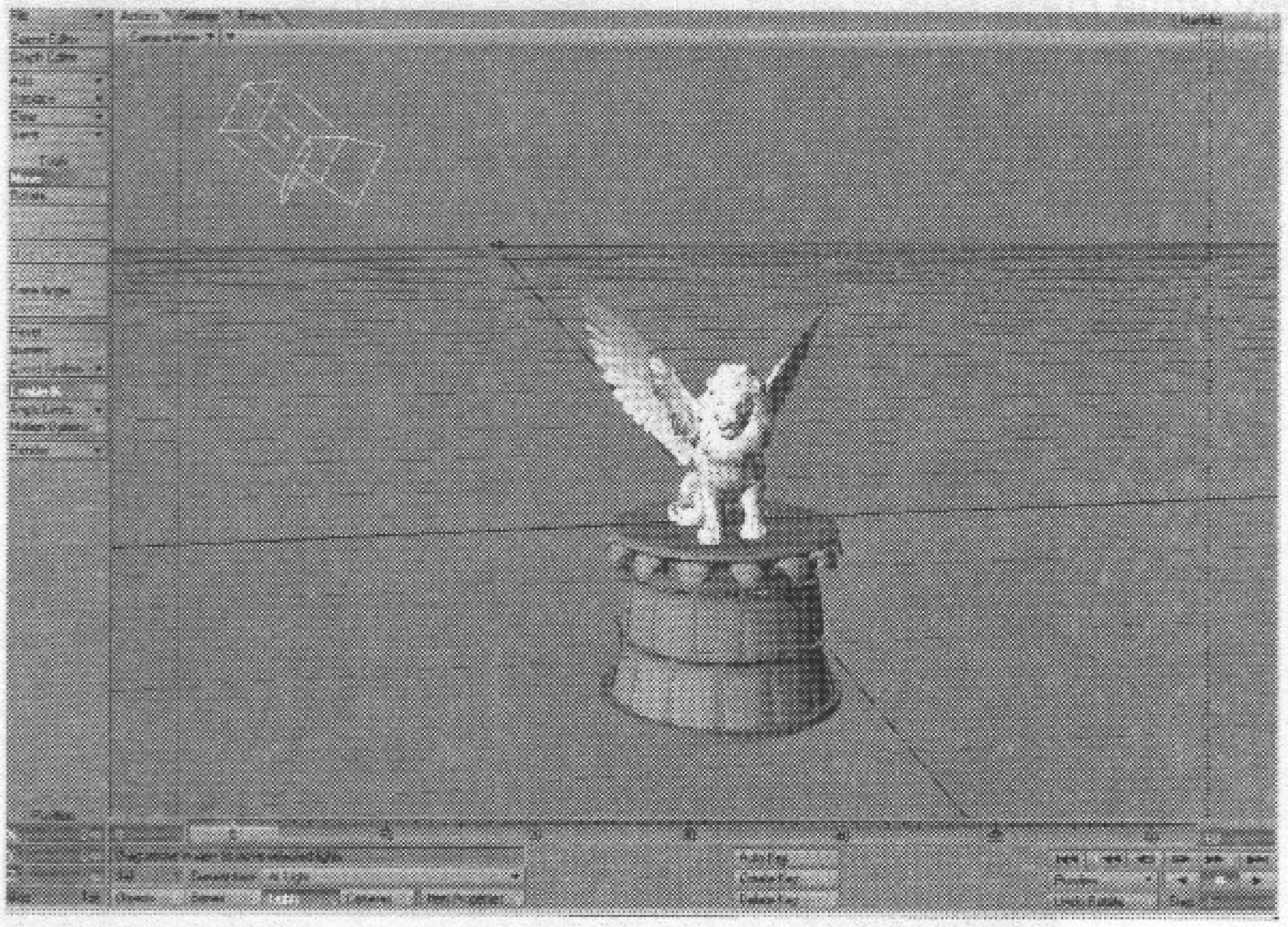
The first thing to remember is that the default Scene loads with one Distant light at 100% intensity and the Ambient intensity set to 25%. This allows you to set up and see your scene and to do test renders.

While this lighting set up is sufficient to work with it really doesn't give an image any depth. Also remember that Ray Trace Shadows, Reflection and Refraction are not turned on in a default scene. Until these are turned on you will not be able to create a photorealistic image. Also with out Ray Trace shadows turned on your objects will not cast shadows on other objects around them.

1 For this tutorial, load the LION STATUE from the TUTORIALS\LIGHTING folder.

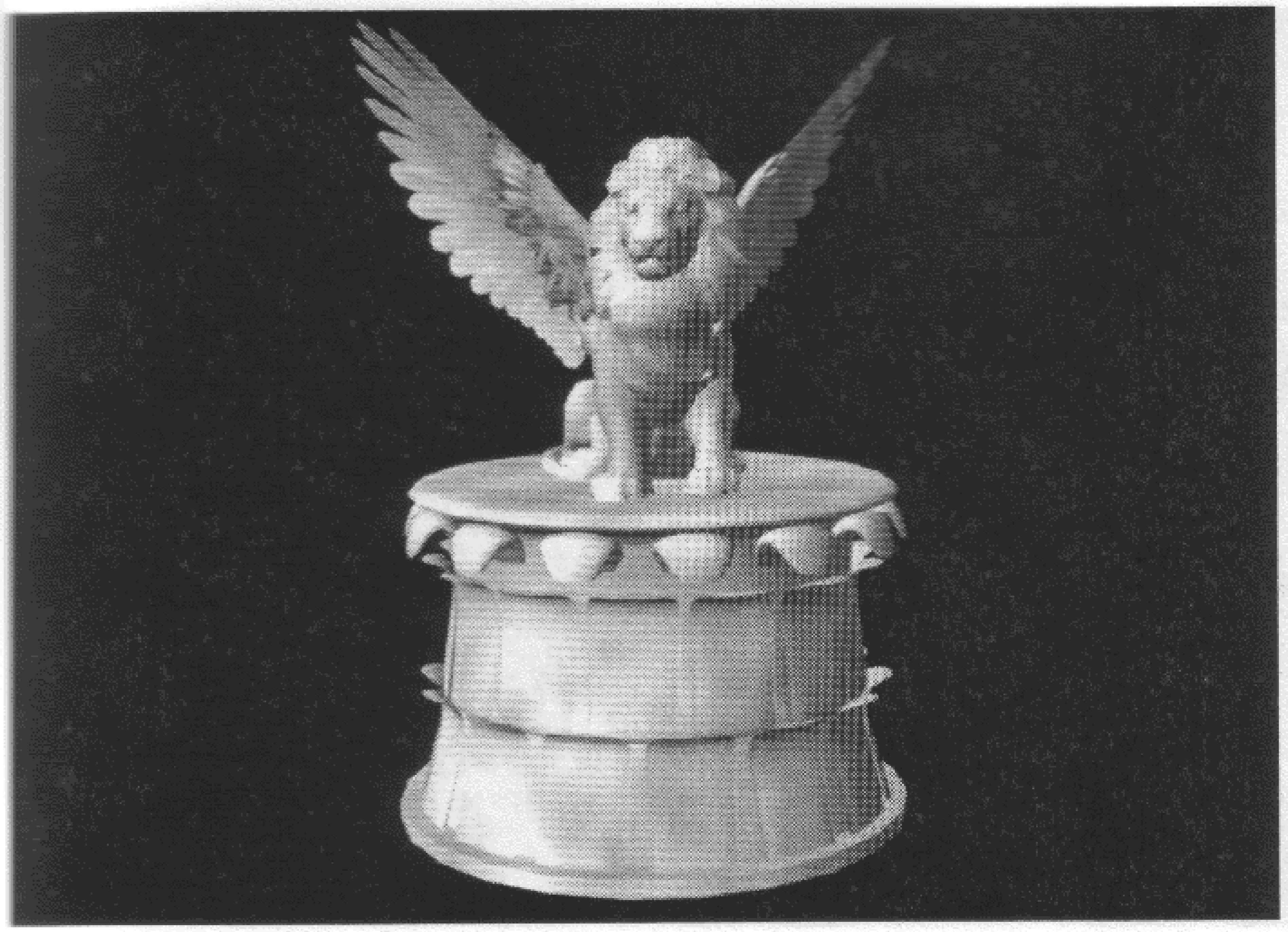
2 Position the camera so you can see the object from a little above and to the left.

3 Move the default light that's already in the scene to the left of the object. This will be our *key* light. Remember to keyframe the new light and camera position or leave **Autokey Create** on in the Global Options panel.



A Default lighting setup

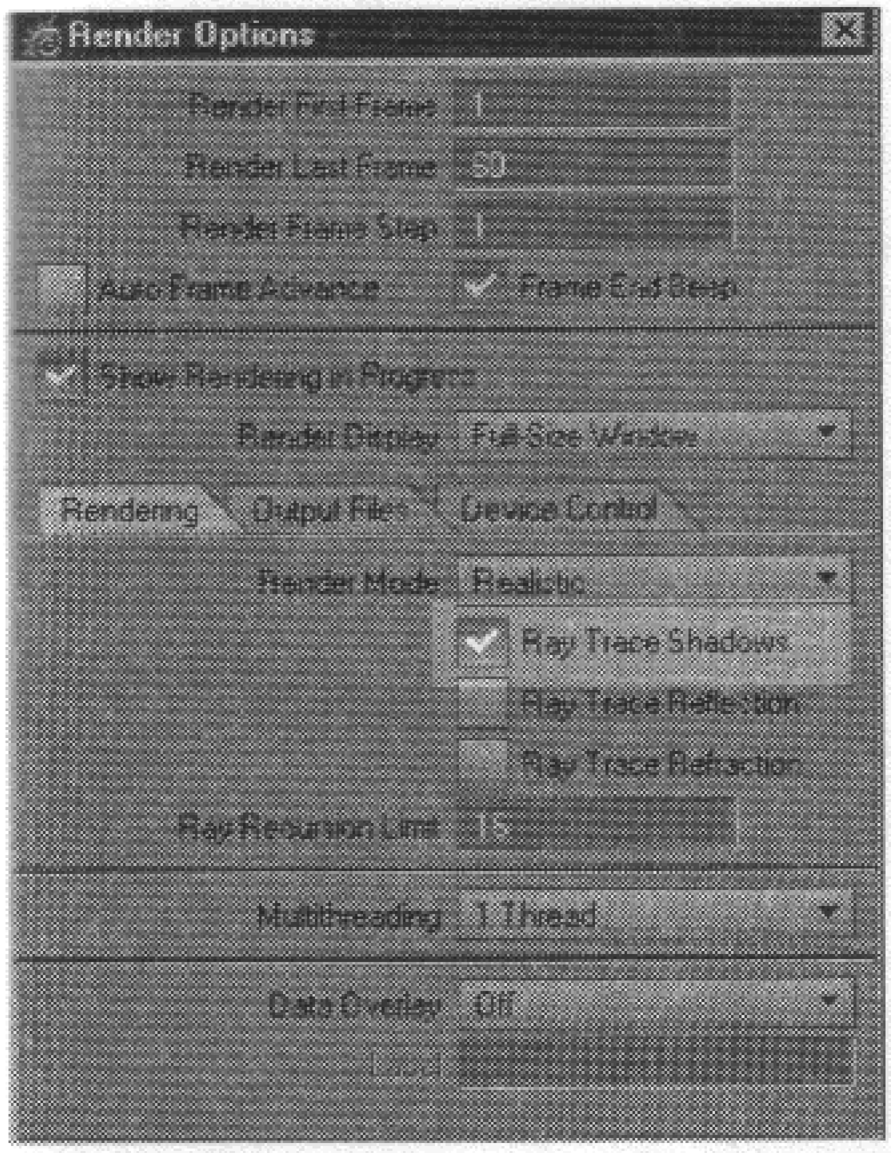
4 Now go ahead and hit F9 to render a test image.



Notice that the lion statue seems to cast a shadow for itself , but it does not cast a shadow on its base.

Note that the image has no depth or contrast.

5 The first thing we need to do is go to the Render options panel (**Actions** > **Render** > **Render Options**) and turn on **Ray Trace Shadows.**



Render Options

Render First Frame 1

Render Last Frame 50

Render Frame Step 1

□ Auto Frame Advance ☑ Frame End Beep

☑ Show Rendering in Progress

Render Display Full Size Window

Rendering | Output Files | Device Control

Render Mode Realistic

☑ Ray Trace Shadows

□ Ray Trace Reflection

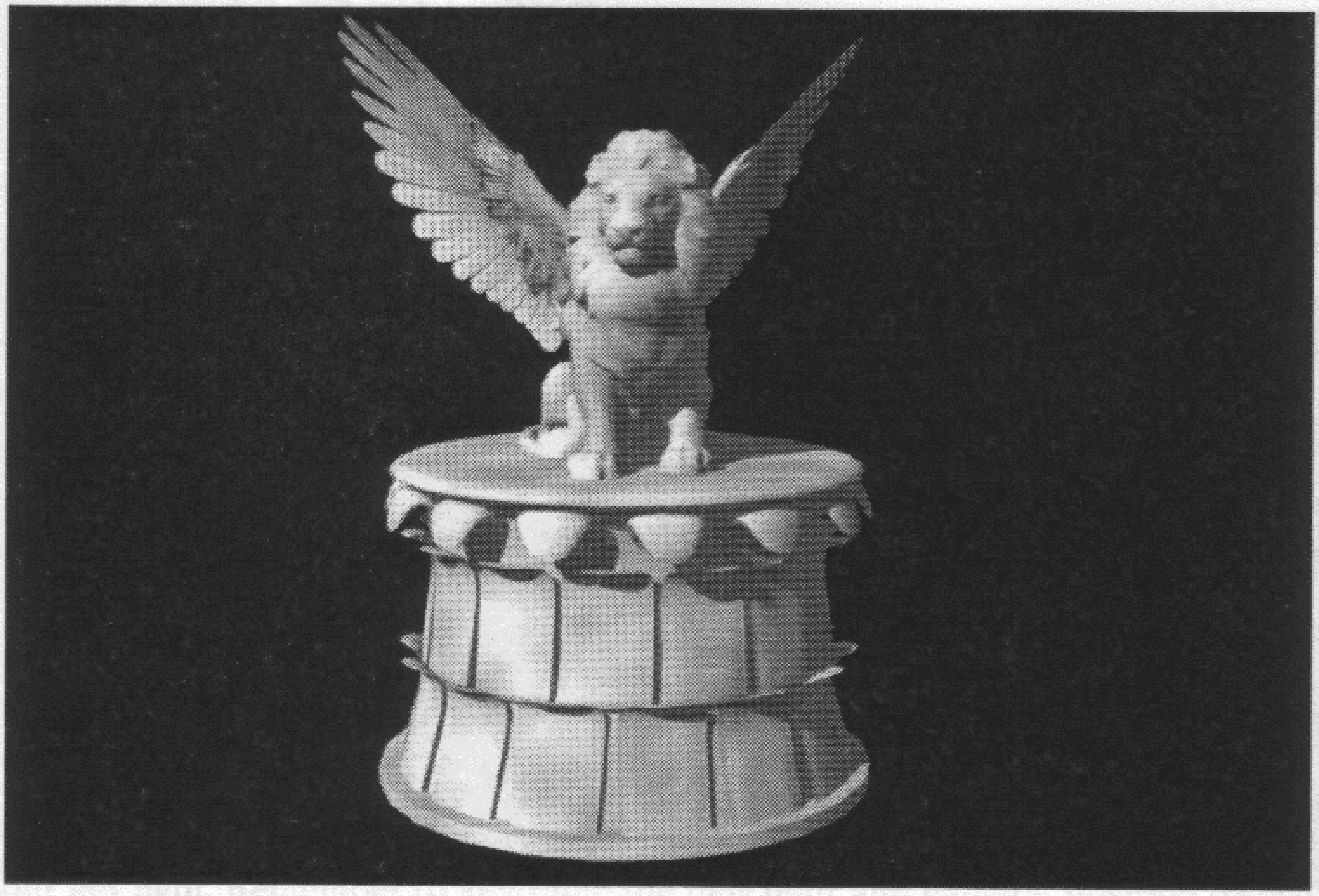
□ Ray Trace Refraction

Ray Recursion Limit 16

Multithreading 1 Thread

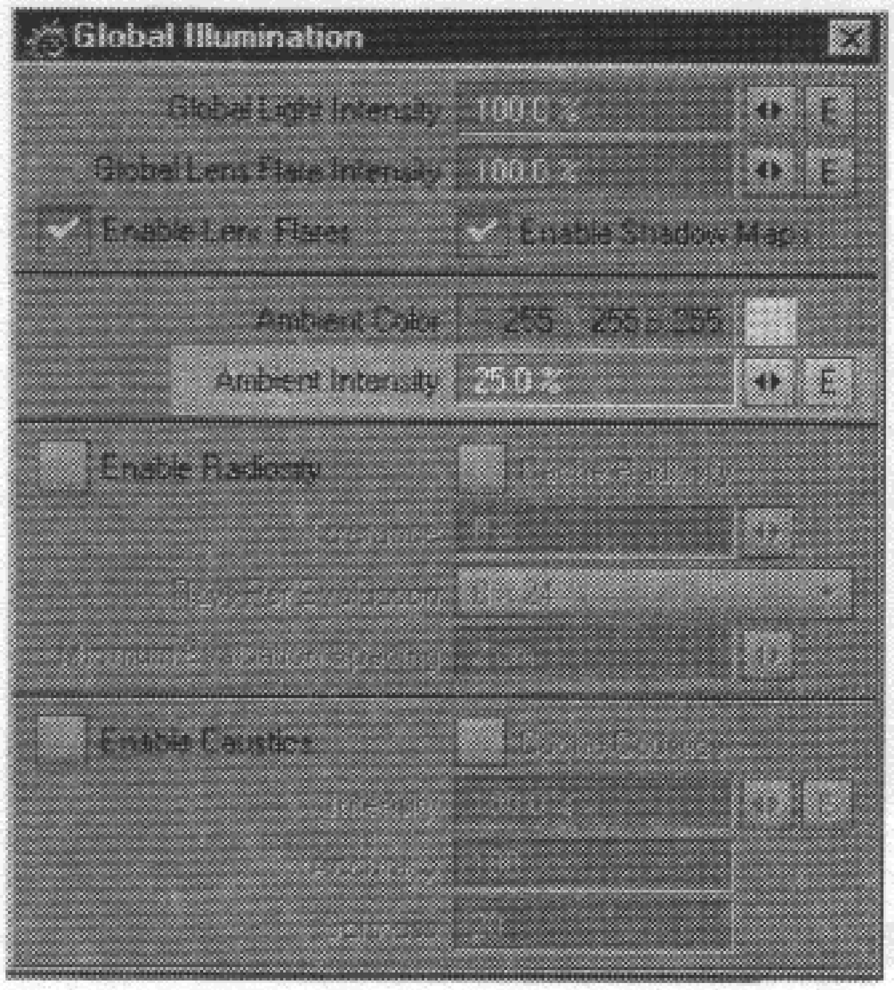
Data Overlay Off

6 Do another test render. This time the lion statue casts a shadow on the base.



But we still have a problem of no real depth to the image. This is because the **Ambient Intensity** is not allowing the shadows to become dark enough.

7 Go to the Global Illumination panel (**Settings** > **Global Illum**). The **Ambient Intensity** is set to the default of 25%. While this can be helpful most of the time, in this case, it's interfering with setting up our lights.



Global Illumination

Global Light Intensity 100.0 %

Global Lens Flare Intensity 100.0 %

☑ Enable Lens Flares ☑ Enable Shadow Maps

Ambient Color 255 255 255

Ambient Intensity 25.0 %

☐ Enable Radiosity

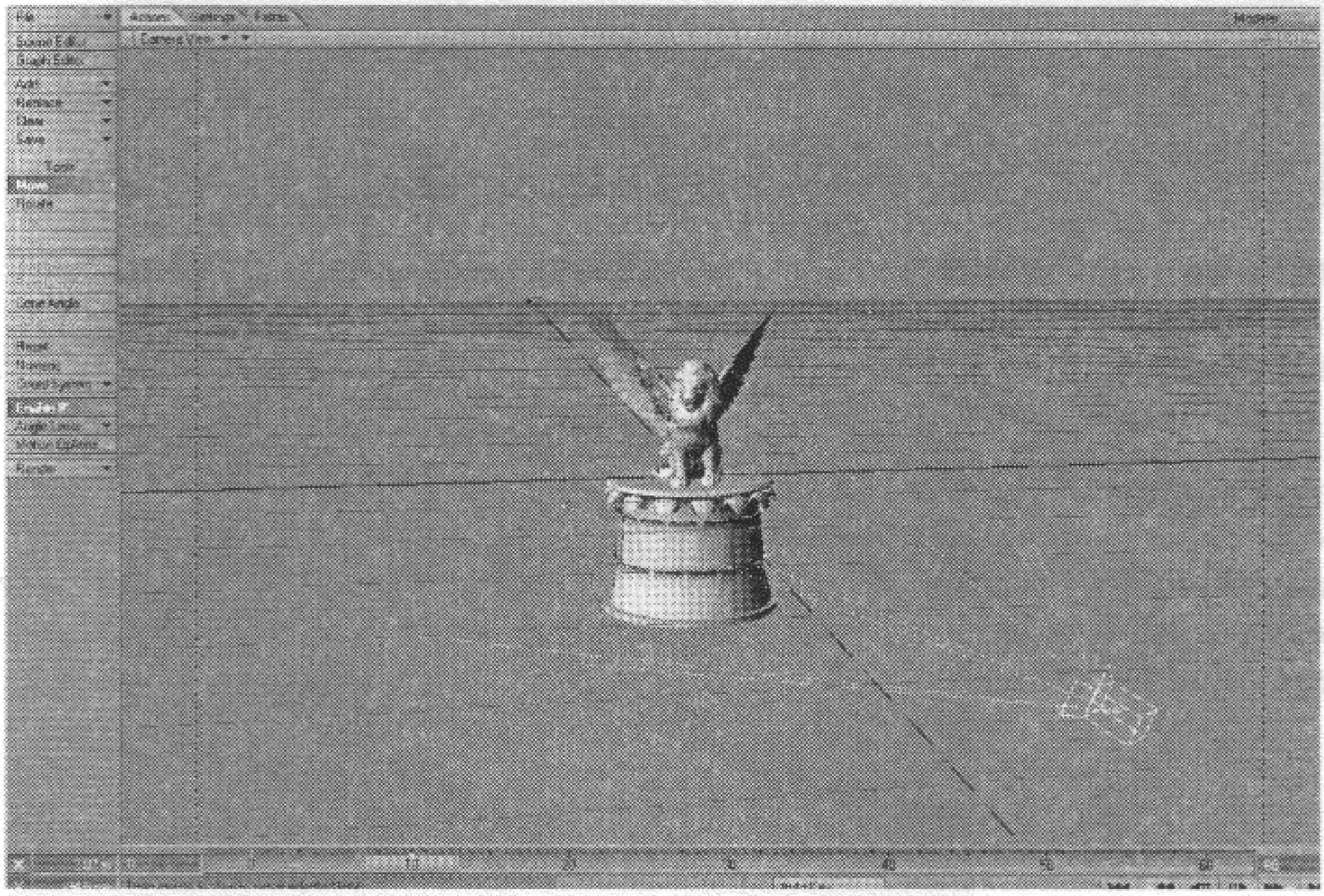
☐ Enable Caustics

8 Turn the **Ambient Intensity** down to 0% and do another test render. Now the image will have black shadows and some contrast and depth to it.

But we still need to see the right side of the statue, which now is in dark shadows.

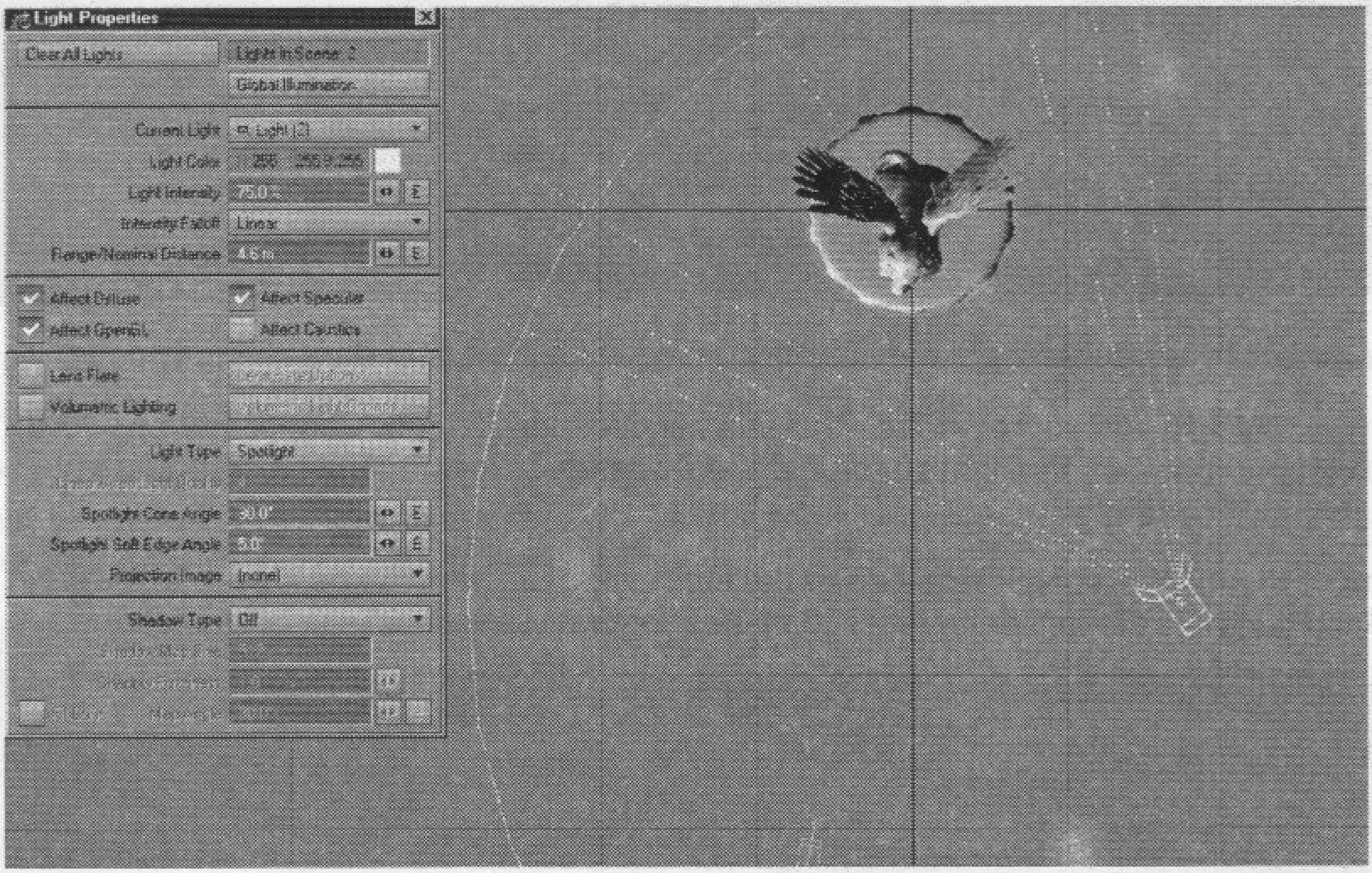
9 Here we need to add another light. This will become our fill light. Choose **Actions** > **Add** > **Spotlight** and position the light to the right and slightly below the object so that it's looking up at the statue.

Remember in any viewport you can go to the Light view and you will be able to see where the selected light is aiming. Remember to make a keyframe for your new light position.



10 Do another test render. This time our shadows are filled in, but it's a little overdone. Whenever you add a new light, its default **Intensity** is set to 50%. In this case, we can do two things, first bring the **Intensity** of the light down or set a falloff distance for the light. Open up the Light Properties panel and select the second light. Normally in Lightwave, a light is calculated at the same intensity along the entire path that the light is projected. But In the real world light falls off over distance.

To recreate this effect in Lightwave, you can set a light to falloff at a certain distance. Set the **Intensity Falloff** to **Linear**. Now in one of your viewports, go to the Top view and adjust it so you see your light. There should be a circle around the light. This is the edge of where the light is falling off at. If you adjust the **Range/Nominal Distance** in the Light Properties panel, you will see the circle expand or shrink. Adjust it so that it reaches just a little past the statue. Also set the **Light Intensity** to 75%.


Light Properties

Clear All Lights | Lights in Scene: 2
Global Illumination...

Current Light: Light (2)
Light Color: 255 255 255
Light Intensity: 75.0 %
Intensity Falloff: Linear
Range/Nominal Distance: 4.5 m

☑ Affect Diffuse ☑ Affect Specular
☑ Affect OpenGL □ Affect Caustics

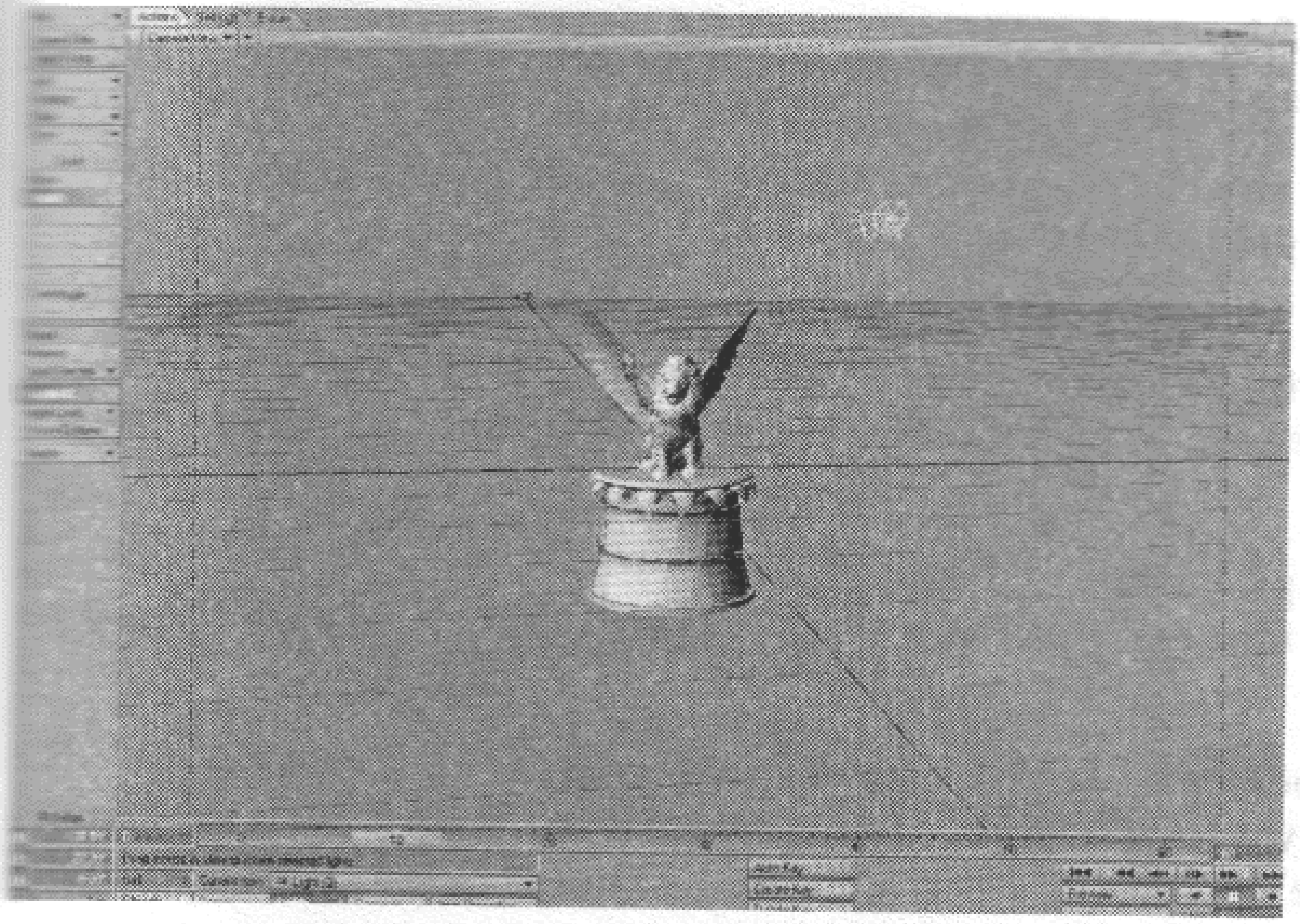
□ Lens Flare
□ Volumetric Lighting

Light Type: Spotlight
Spotlight Cone Angle: 30.0°
Spotlight Soft Edge Angle: 5.0°
Projection Image: (none)

Shadow Type: Off

When you do a test render this time, the areas with dark shadows should be filled in. You want it so the shadows are still there but just enough so that you can see some details. Change the **Light Intensity** and the **Range/Nominal Distance** and do test renders to see the effect both have on the scene.

11 Now that we have our main light and a fill light, we want to add one more. This will be a *highlight*. Add another shadow-mapped spotlight and place it behind and above the statue. Again, set the **Range/Nominal Distance** to just past the statue and make it's **Light Intensity** at 50%. This throws a little bit of light on the shoulders and other high points of the statue making those areas stand out more.



12 To add a little variety, change the colors of the lights so they have a slight color tint to them.

13 Now we need to cover one last thing and that is volumetrics. Volumetrics are not lights but a lighting effect. This is the effect you get when shining light through fog or smoke.

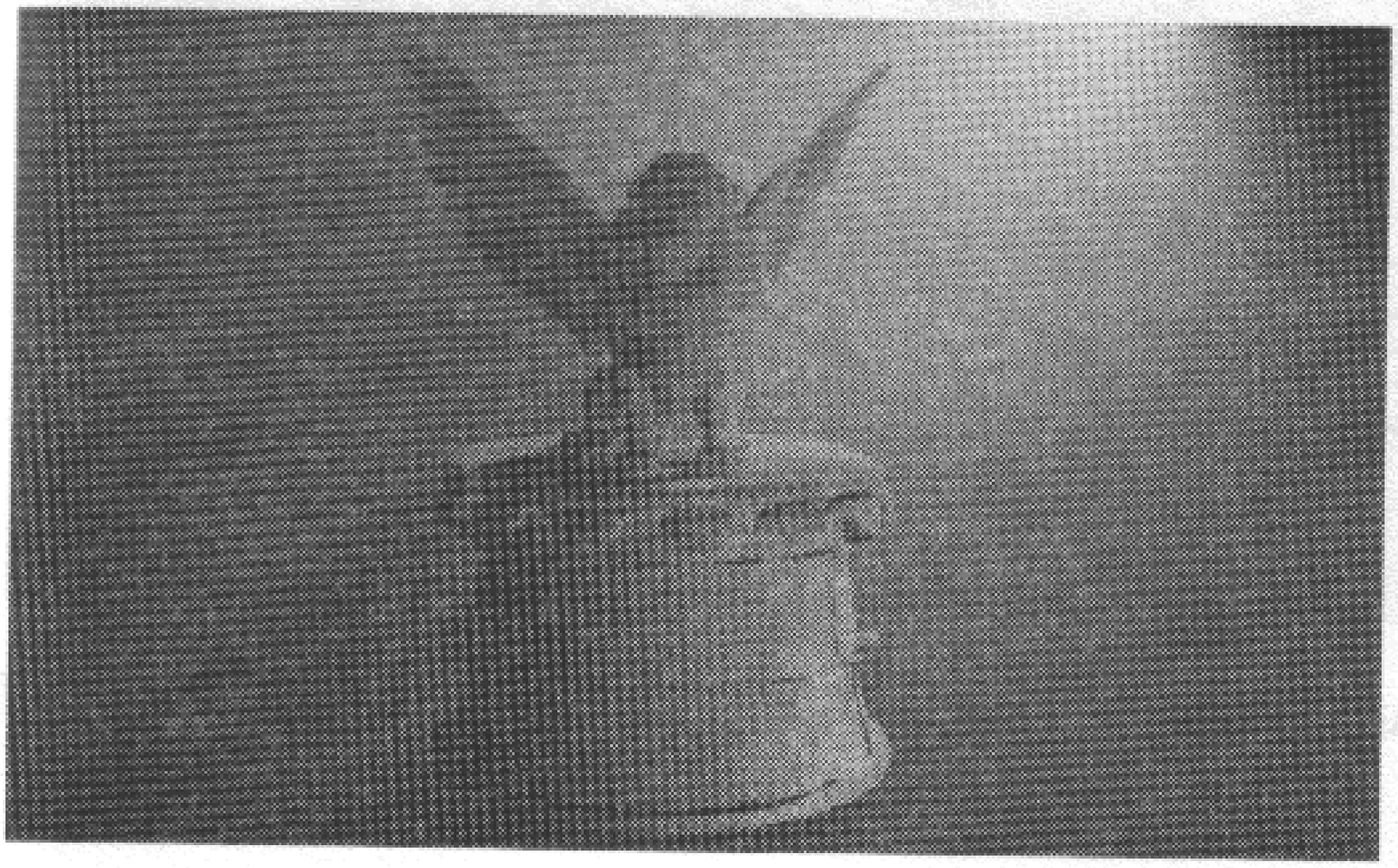
14 Setting up Volumetric lights is very easy. First, pick which light you want to use as the volumetric light source. Volumetric lighting works best with shadow-mapped lights.

15 In the Light Properties panel you will see **Volumetric Lighting**, turn it on by clicking in the box next to it.

16 Then go into the Volumetric Light Option panel. The three main things you want to be concerned with are **Quality**, **Height** and **Luminosity**.

17 Set the **Quality** to **Low** for test renders and **Good** to **Medium** for the final render.

18 If you do a test render and see only black, increase the **Height**. **Height** is basically how far the effect extends outward from the light source. **Luminosity** is how bright the effect is. If you do a test render and see nothing but white decrease the **Luminosity**.



UV MAPPING TUTORIAL

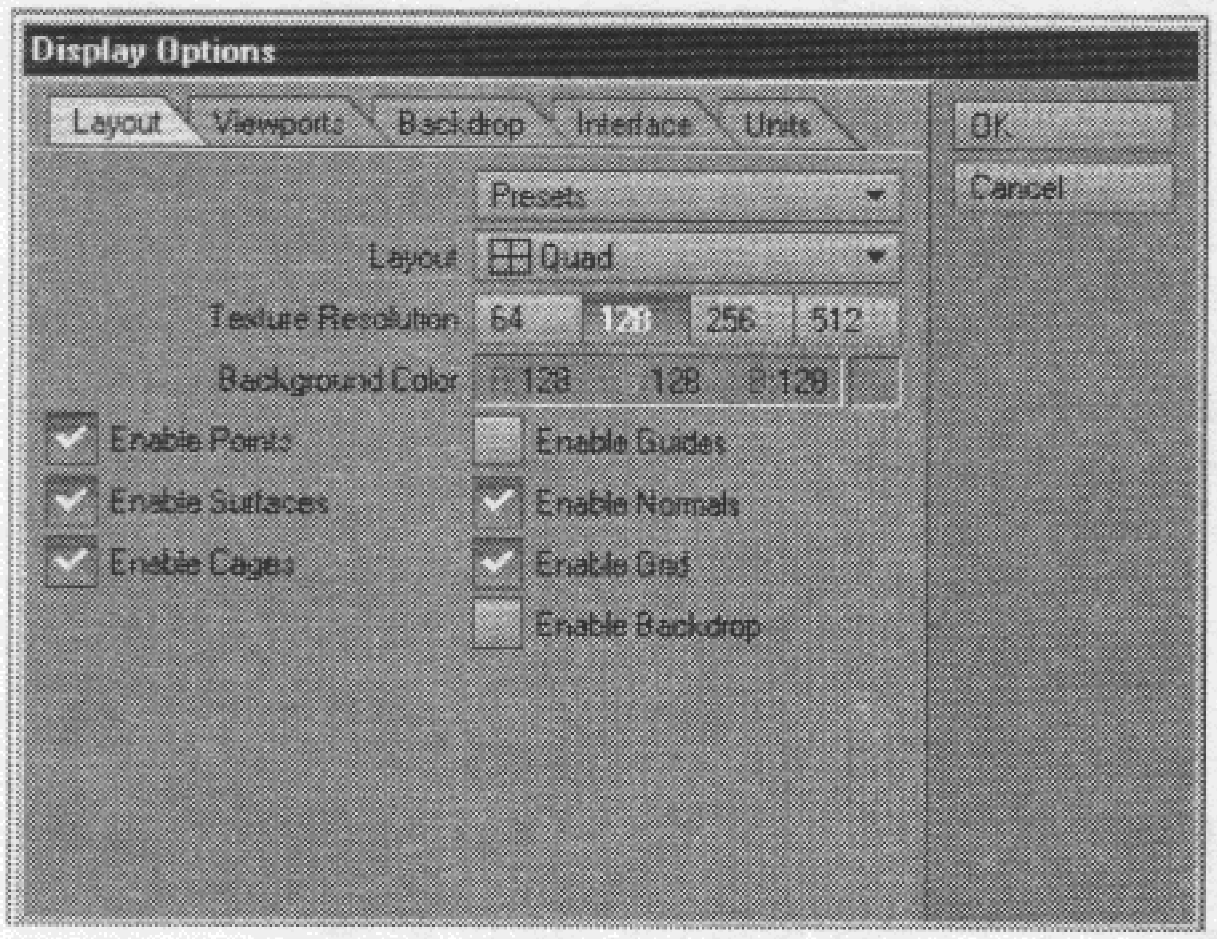
One of the most important aspects of any object is texture mapping.

Texture mapping helps make an object look more realistic. Texture mapping is just applying an image to a selected set of polygons. Imagine that you are really just projecting the image onto the object.

This works fine for objects with well-defined shapes. But let's say, for example, you have a human face. If you just project the image onto the face you will see the image start to tear and smear along the sides.

The only way to correct this problem is by UV mapping. UV mapping uses a set of internal 3D coordinates to adjust the image to fit the object. This is a method that is used a great deal in the gaming industry.

1 Creating an UV map in Modeler is very easy. First, we need to set up the Modeler screens so that we see what we are doing. Now there are several different ways to arrange the Modeler screens. Choose **Display** > **Options** and set the **Layout** to **Quad**.



Layout: Quad

Texture Resolution: 64 | 128 | 256 | 512

Background Color: R 128 G 128 B 128

☑ Enable Points

☑ Enable Surfaces

☑ Enable Cages

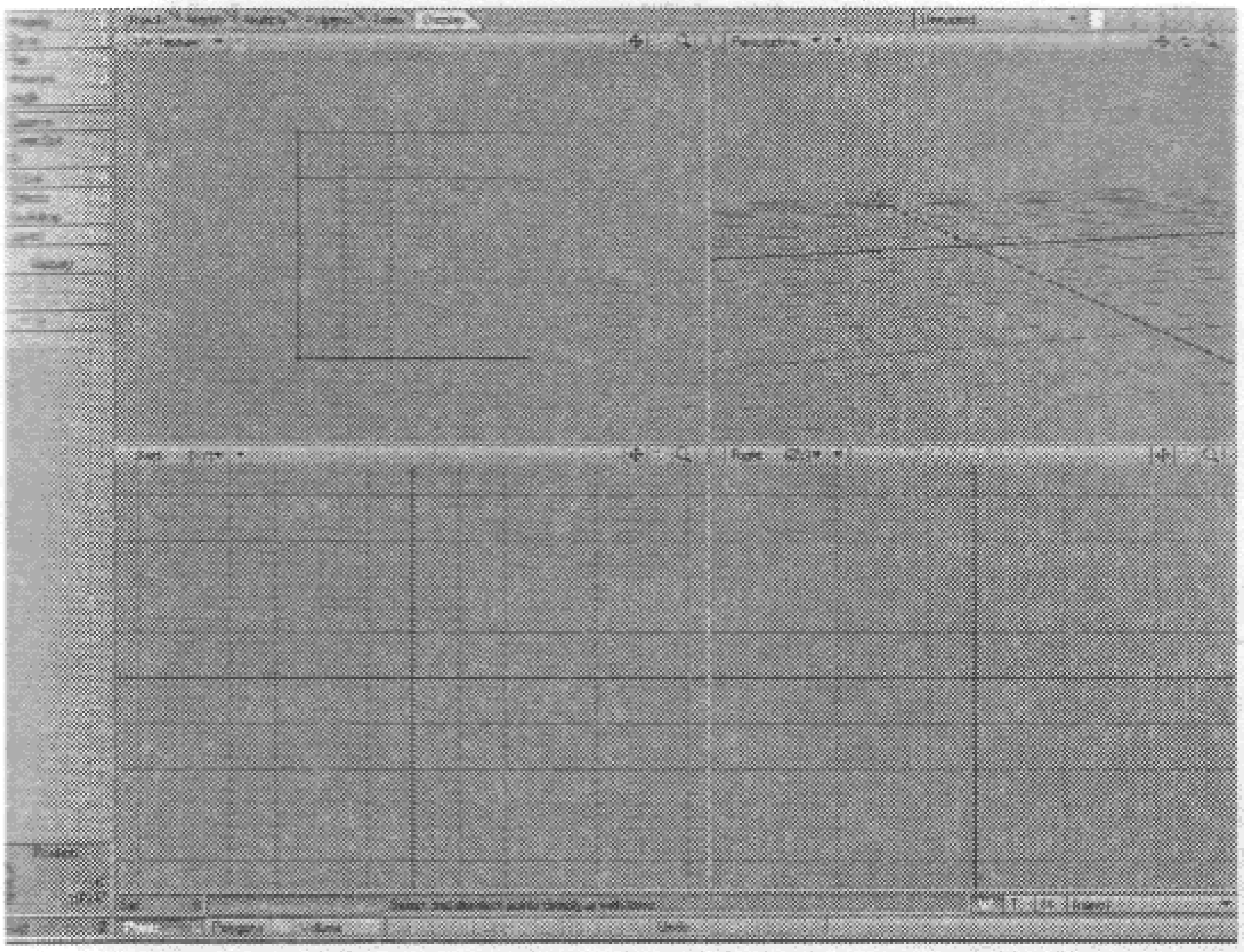
☐ Enable Guides

☑ Enable Normals

☑ Enable Grid

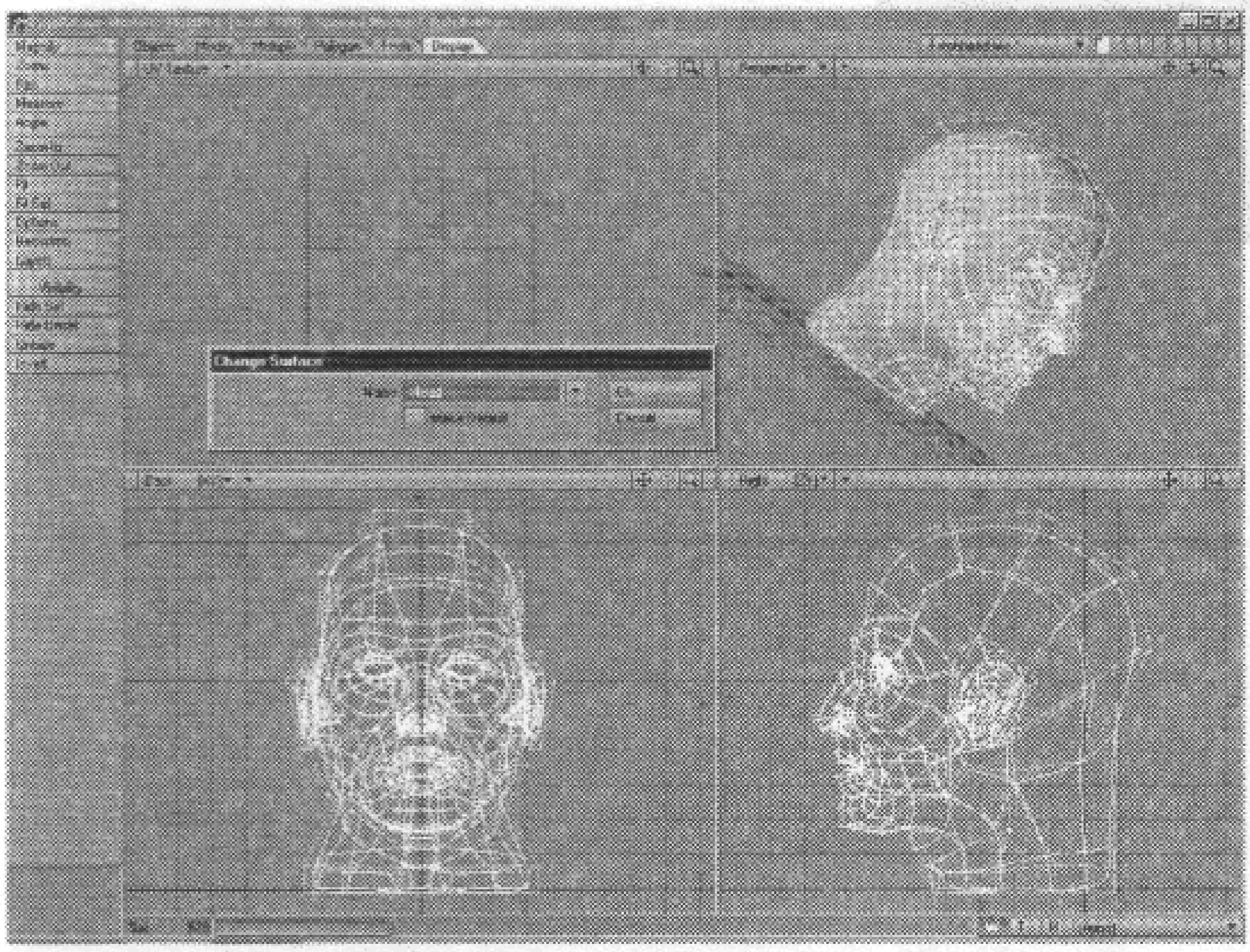
☐ Enable Backdrop

2 Set the Perspective viewport 2's **Rendering Style** to **Texture**. Now in the upper-left-hand screen, viewport 1, set the **View Type** to **UV Texture**. This is the window that will display our UV Map. Make note of the fact that nothing is displayed there yet.



3 Now load the TARON_HEAD object from the TUTORIALS\UV_MAPPING directory. Press the A key to center the object on all viewports. The first thing we have to do is assign a texture name to the polygons we want to map. In this case we are mapping the entire object. If you just want to do a small area you would just use the normal selection tools to select the polygons you want to name. In either case, press the SPACEBAR until **Polygon** (edit mode) at the bottom of the modeler interface is highlighted. While holding down the RMB, draw a circle around the object in any screen. This selects all of the polygons for the object.

4 Now press the Q key, this will bring up the surface naming panel. Uncheck the make default button and give the object a surface name of something like HEAD. Now we have a set of polygons to receive our UV map.



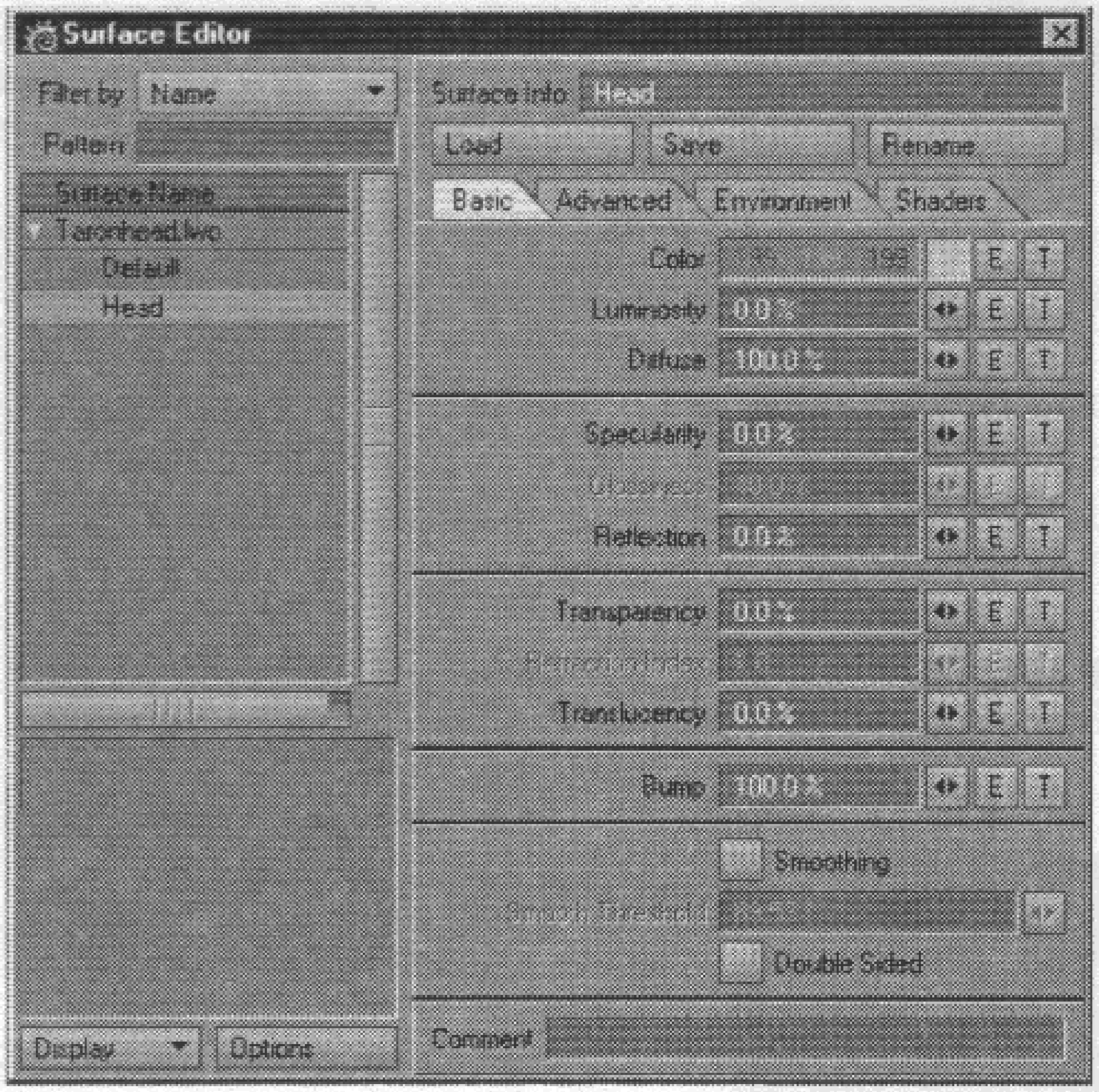
Next we need to tell Modeler that we are going to be making a UV map.

5 Choose **Tools** > **Make UVs**. This will bring up the Assign UV Coordinates panel. Now we get to the tricky part. There are three different ways of making the map: planar, cylindrical and spherical. And each of these can be projected on the X, Y or Z axis.

6 In our case, we want to set the **Map Type** to **Cylindrical** and the **Axis** to **Y**. Use HEAD as the (UV) **Texture Name**. Imagine that you're holding a can in front of you and you're wrapping a label around it. Now click **OK** and look in the UV mapping window.

7 Now we have an image of our object but it's laid out flat.

8 What you're basically doing here is unwrapping the object and laying it flat. Imagine taking the object, unstitching it from the back and laying it flat on a table. Note that this window only displays UV coordinates. There are points and polygons in this window but changing them has no effect on your object.



Filter by: Name

Surface info: Head

Surface Name: Taronhead.lwo, Default, Head

Color: 198

Luminosity: 0.0 %

Diffuse: 100.0 %

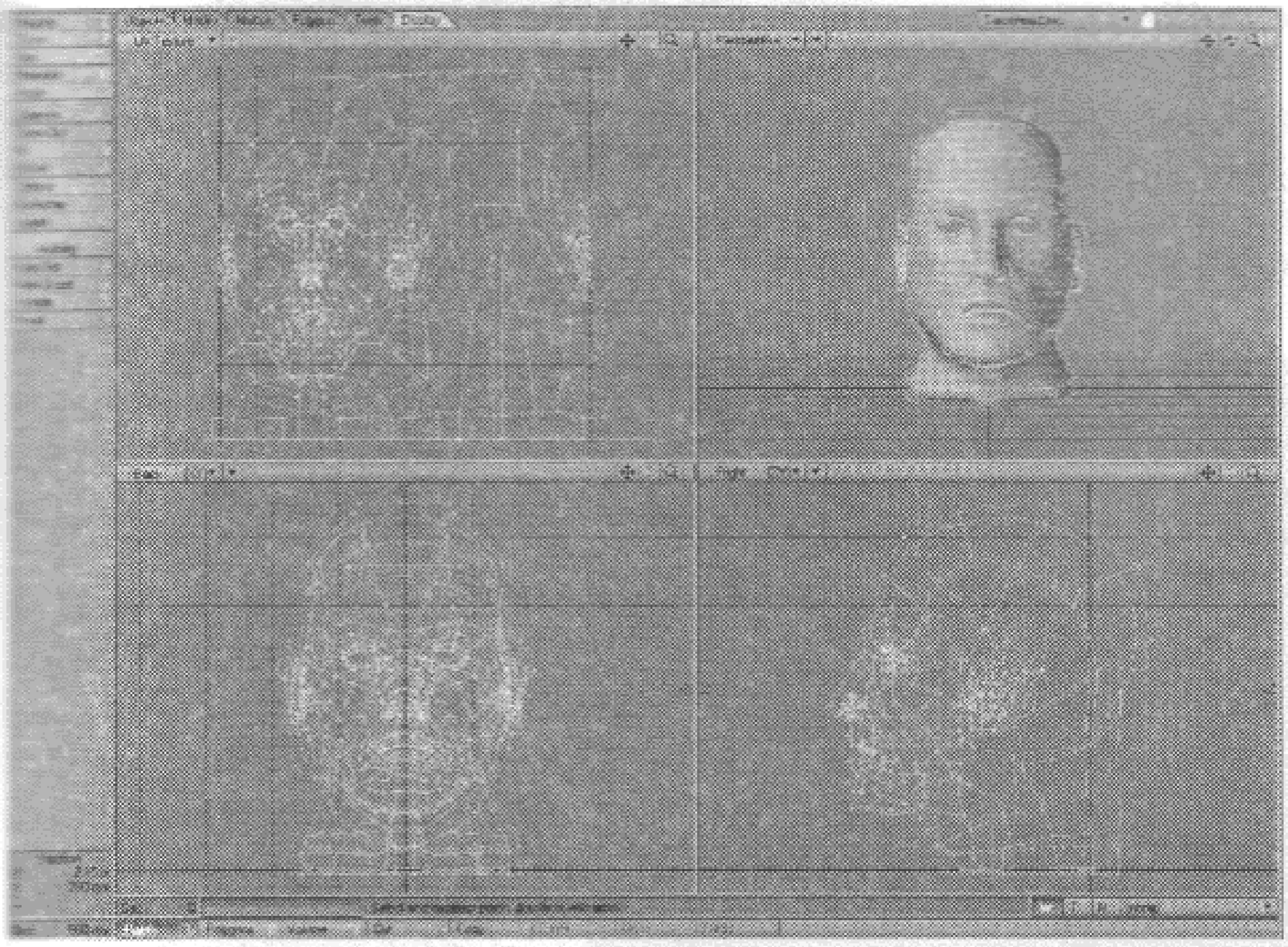
Specularity: 0.0 %

Reflection: 0.0 %

Transparency: 0.0 %

Translucency: 0.0 %

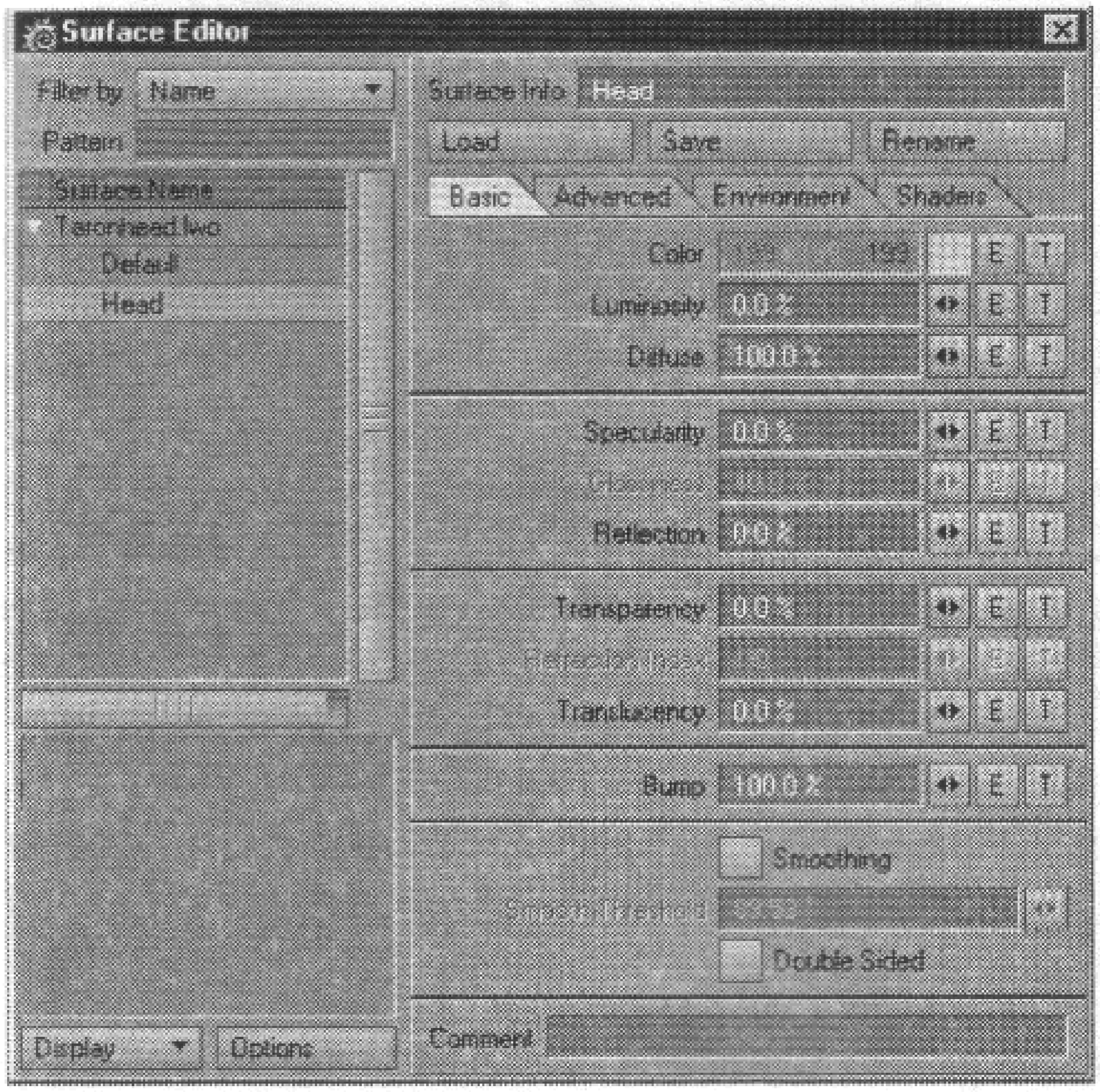
Bump: 100.0 %



9 Now we have to load and assign our image map that we are going to be applying. Open the Surface Editor. Under the surface name listing, click on the small white arrow next to Taron_head.lwo. This will display all of the surface names for our object.

10 In this case, we only have one so select it. Click on the **Color** Texture button. This will open the Texture Options panel.

11 Now look on the right side and set the **Projection** to **UV**. Next, set **UVmap** to Head. Next, select **(load image)** from the **Image** pop-up menu. Load the TUTORIALS\UV_MAPPING\FACE image. Now look down towards the bottom of the panel and click on **Automatic Sizing**. This will help get the image roughly aligned to our surface.



Surface Editor

Filter by: Name

Pattern

Surface Name
- Taronhead.lwo
  - Default
  - Head

Surface Info: Head

Load | Save | Rename

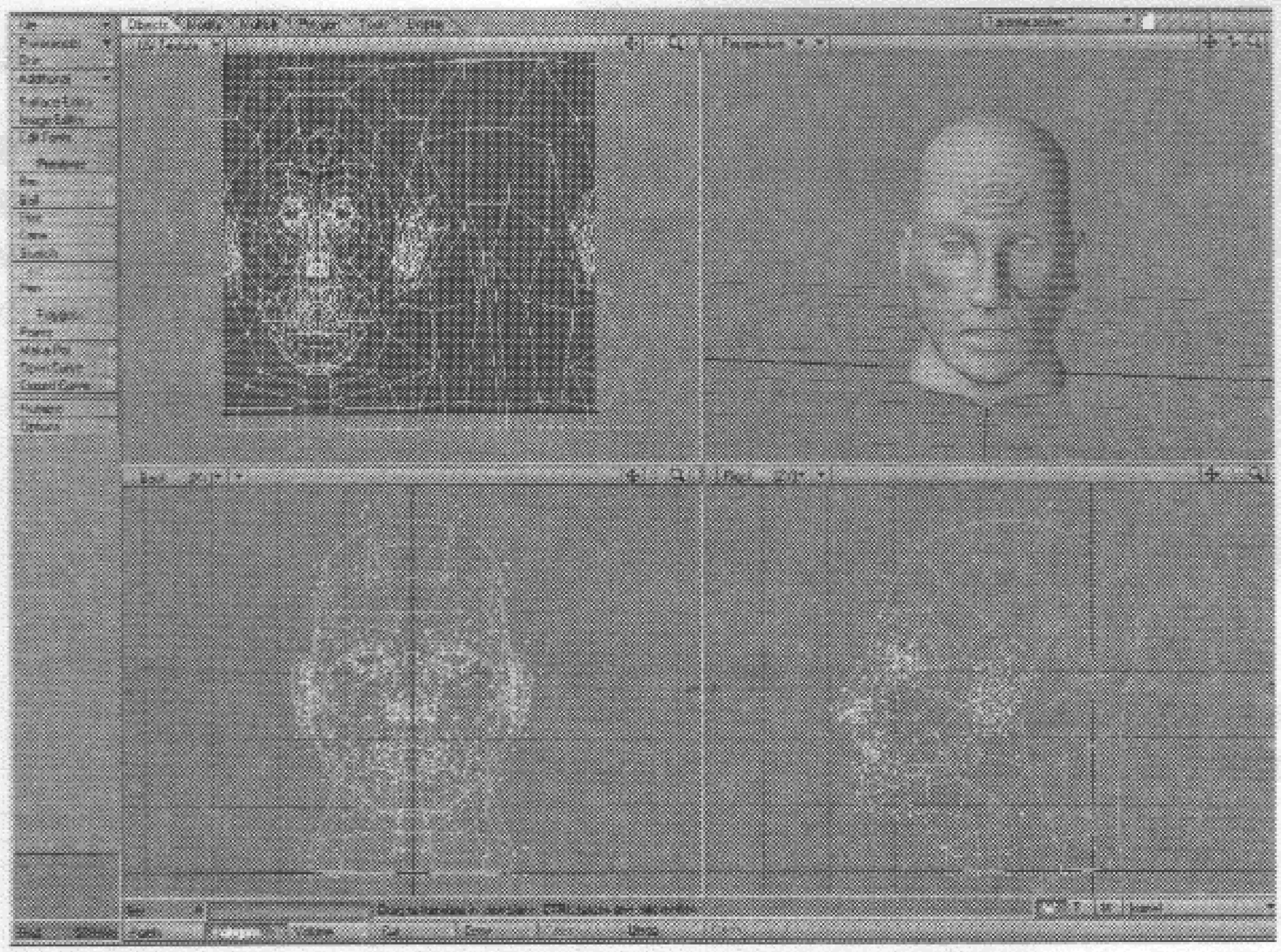
Basic | Advanced | Environment | Shaders

Color: 199 199 E T
Luminosity: 0.0 % E T
Diffuse: 100.0 % E T
Specularity: 0.0 % E T
Glossiness
Reflection: 0.0 % E T
Transparency: 0.0 % E T
Refraction Index
Translucency: 0.0 % E T
Bump: 100.0 % E T
□ Smoothing
Smoothing Threshold
□ Double Sided

Comment

Display | Options

12 You should see your object with the texture map applied in the Perspective viewport. It's probably a little off though. This is where the UV mapping window comes into play. Choose Display > Backdrop to open the Display Options panel, Backdrop tab. Click on **Viewport** 1 and select the Face image on the **Image** pop-up menu. Click **OK** to close the window. Now in our UV mapping window, we should see the face texture behind our unwrapped face.



13 At this point, you can use most of the normal modeling tools to move the points and polygons in the UV window, so that it aligns with your texture map.

14 Notice that as you move points and polygons around in the UV window that it seems to keep wrapping around. This is because the image you are seeing is just the object laid out flat. Thus, as you move the polygons around in the UV mapping window it will try and wrap the image completely around the object.

Now one question that you may have is what is the best way to make image maps for UV objects. Well, right after you make the UV texture map assignment listed above, stretch the screen out so that the UV mapping window fills your screen. Using either your PRINT SCRN key or screen capturing software and grab an image of your screen. Then load that image into your paint program and use it as a painting guide.

MORPH MAPPING AND AUDIO

In this tutorial we will cover *morph mapping*. Morph mapping allows you to change features of a model without having to build or load another model. This is mostly used for character animation. For example, if you wanted to have a character talk, you could use bones, but you would have to use a lot of bones in the face in order to get the expressions you wanted.

Morph mapping allows you to get a wide range of character expressions without having to set up complicated bones. It's also very easy to animate.

1 First open up Modeler and load the TARON_HEAD object from the TUTORIALS\UV_MAPPING directory.

2 If the object looks like a mass of polygons, hit the TAB key to activate the SubPatch mode. Notice how the object changes and smoothes out. SubPatch allows us to build a fairly complicated model with a few polygons. This will also make it easier for us to change the model.

3 First we need to set things up so that we can see them. Choose **Display** > **Options** and turn off **Grid** and **Cages**. This will lets us see what we are working on more clearly. Also make sure your Perspective window is set to **Smooth Shade**.

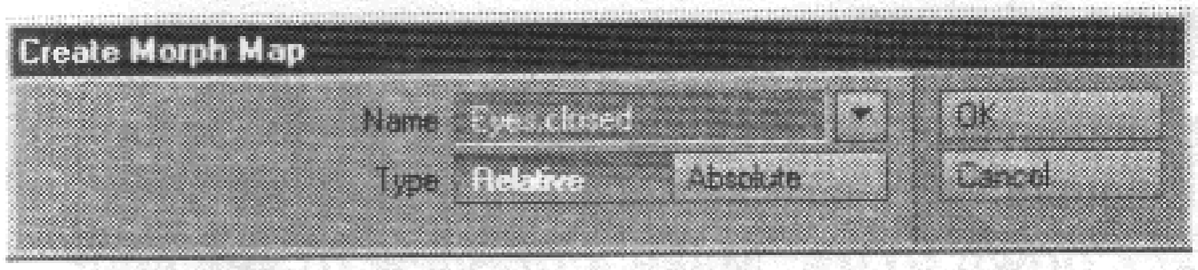
4 Now we need to understand the relationship between our objects and our changes. The object that we have now will be called our *base object*. The base object is the one we will load into layout. It is also the object that we will *base* all of our changes off of. The changes that we make to the base object will be our morph maps. To better understand this, let's make a simple change.

5 Look in the lower right hand corner of Modeler. You will see three letters: **W**, **T** and **M**. These stand for weight, texture and morph vertex maps (aka VMaps).

6 Click on the **M**. We will now be able to make new morph maps for our object. To the right of the **M** you will see **(base)**. This is a pop-up menu for selecting maps.

7 Choose **(new)** from the pop-up. A panel will appear allowing us to name our new morph map. We are going to call this one EYES.CLOSED. The reason for the period in the middle of the name is this: when we load this model into Layout and use Morphmixer, it will create a tab on the panel for us called *Eyes*. This will help us organize our maps better. But we will get to that later.

Also make sure **Type** is set to **Relative**. This means the changes are *relative* to the base object. You would use **Absolute** when the entire object changes shape, for example, form a box to a ball.


Create Morph Map

Name Eyes.closed

Type Relative Absolute

8 Zoom in on the head until you can see the eyes and hit the SPACEBAR until you are in Point Selection mode. What we want to do is select just the points on the edge of the eye lids. Once you have those points selected use the **Drag** tool to bring them down like they are closed.

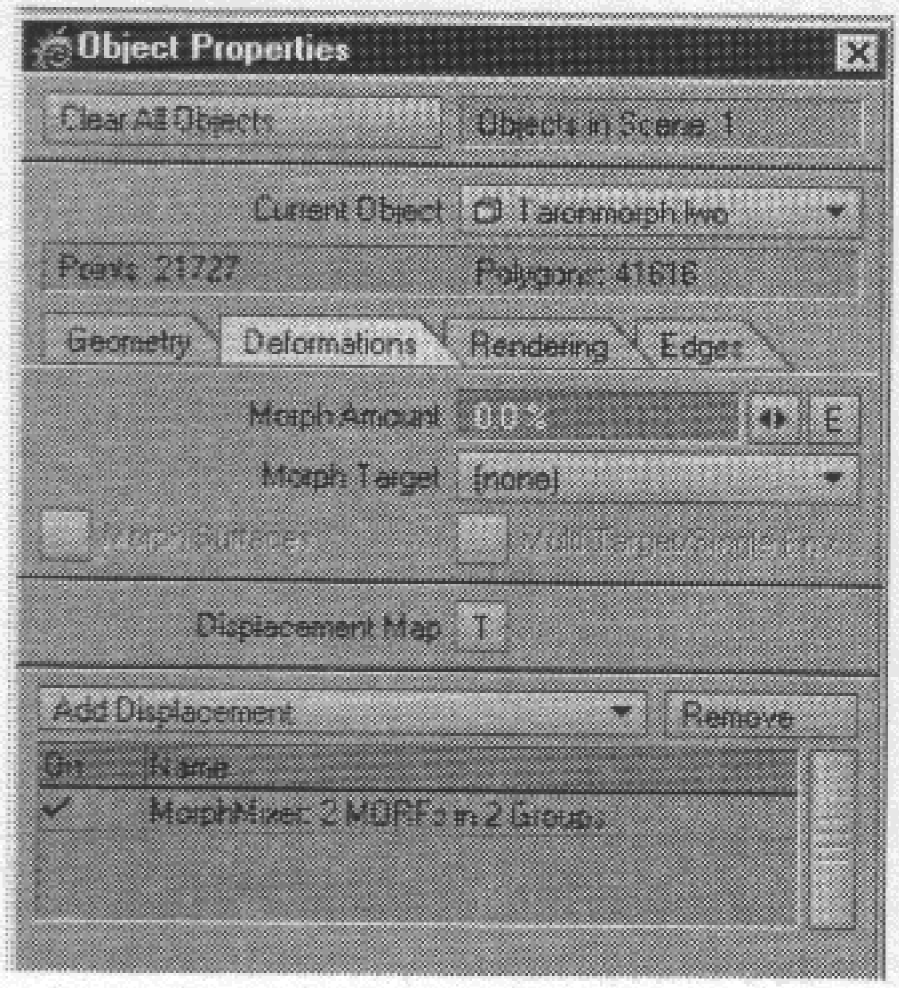
9 Go back down to the morph map pop-up menu and go back up to base. Notice that the eye lids go back up to where they were before. If you go back to the pop-up menu and go to eyes.closed you will see the eye lids closed.

10 Now we have our first morph map done. In order to create a second one we need to go back to our base object. The reason we want to do this is because all changes are based relative to the base object. Create a new map and call it MOUTH.CLOSED. Now you can either select the points that make up the mouth or select the polygons. Either way you want to make the mouth closed. You can use the **Drag**, **Stretch** and/or **Move** tool to do this. But you

cannot add or subtract polygons or points from the model. Keep this in mind when you are creating your model.

11 For now let's keep our changes to just these two. Save the model as TARONMORPH. Also minimize Modeler and start Layout. If the Hub is running, the object will appear automatically in layout. Otherwise, load the saved object. By the way, an object with morph maps is called an EndoMorph object.

12 Open the Object Properties panel and select the Deformations tab. Select MorphMixer from the Add Displacement pop-up menu.



Objects in Scene: 1

Current Object: Taronmorph.lwo

Points: 21727

Polygons: 41616

Morph Amount: 0.0 %

Morph Target: (none)

On	Name
✓	MorphMixer: 2 MORFs in 2 Groups

13 Open the MorphMixer options panel by double-clicking it in the list. Now you see that we have two tabs; one titled Eyes and another called Mouth. We set this up in modeler by naming the morph maps Eyes.closed and Mouth.closed.

14 The name before the period will be a tab and after that will be a slider name. So if we look at the Eyes tab we will see a closed slider that ranges from 0% to 100%. As you move this slider to 100% you will see the eyes your model close.

15 Animation using the sliders is very simple. Turn on Auto Key Create and go to frame 20. Slide the eyes closed slider to 100% and go to the mouth tab and slide it closed to 50%. Now go to frame 40 and move both sliders back down to 0%. Make a preview to see your animation.

16 Now we have one extra step that we can do to make animating a little easier. Go to the Scene Editor and select Load Audio from the **Audio** pop-up menu. Load the CRUSH.WAV file form the AUDIO\TUTORIALS folder.

17 Now if you look at the frame slider, a purple graph will appear giving you a visual sample of the audio clip you just loaded. Now as you scroll through your scene, you will be able to *see* where the words are being formed in the audio clip. This will make it easier for you to animate the lip and facial movements of your characters.

CHARACTER ANIMATION AND IK

There are two important factors that need to be considered when setting up a character's bone hierarchy. The first consideration is how the character should move. It is important to know as much about this as possible because it can have a very serious impact on how much time you spend in setup. For instance if you know that a character will only be seen from the waist up for the entire length of an animation, there is no need to spend time setting up legs.

The second consideration is how the bones should deform the character. Do you wish the character to appear gelatinous or rigid?

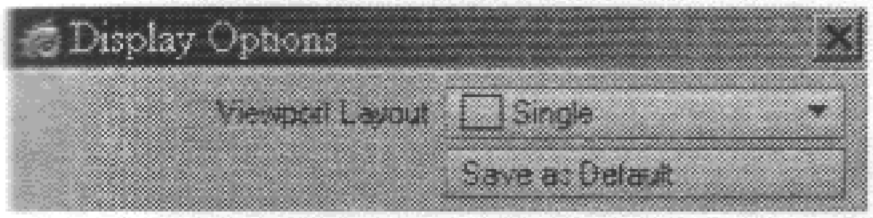
Bone setup plays an important role in how a character's shape changes as it moves. In the past, many users have complained about the difficulty they experienced controlling the shape of boned objects in Lightwave. LightWave [6] has several tools which simplify the process and allow you more control over your final results.

Simple Planar IK Setup

Make the Chain
In this tutorial we will create a fairly simple leg-bone chain.

1 Open Layout.

2 If it isn't already, set the **Viewport Layout** to **Single** (**Extras** > **Display Options** > **Viewport Layout**) then close the Display Options panel.



Display Options

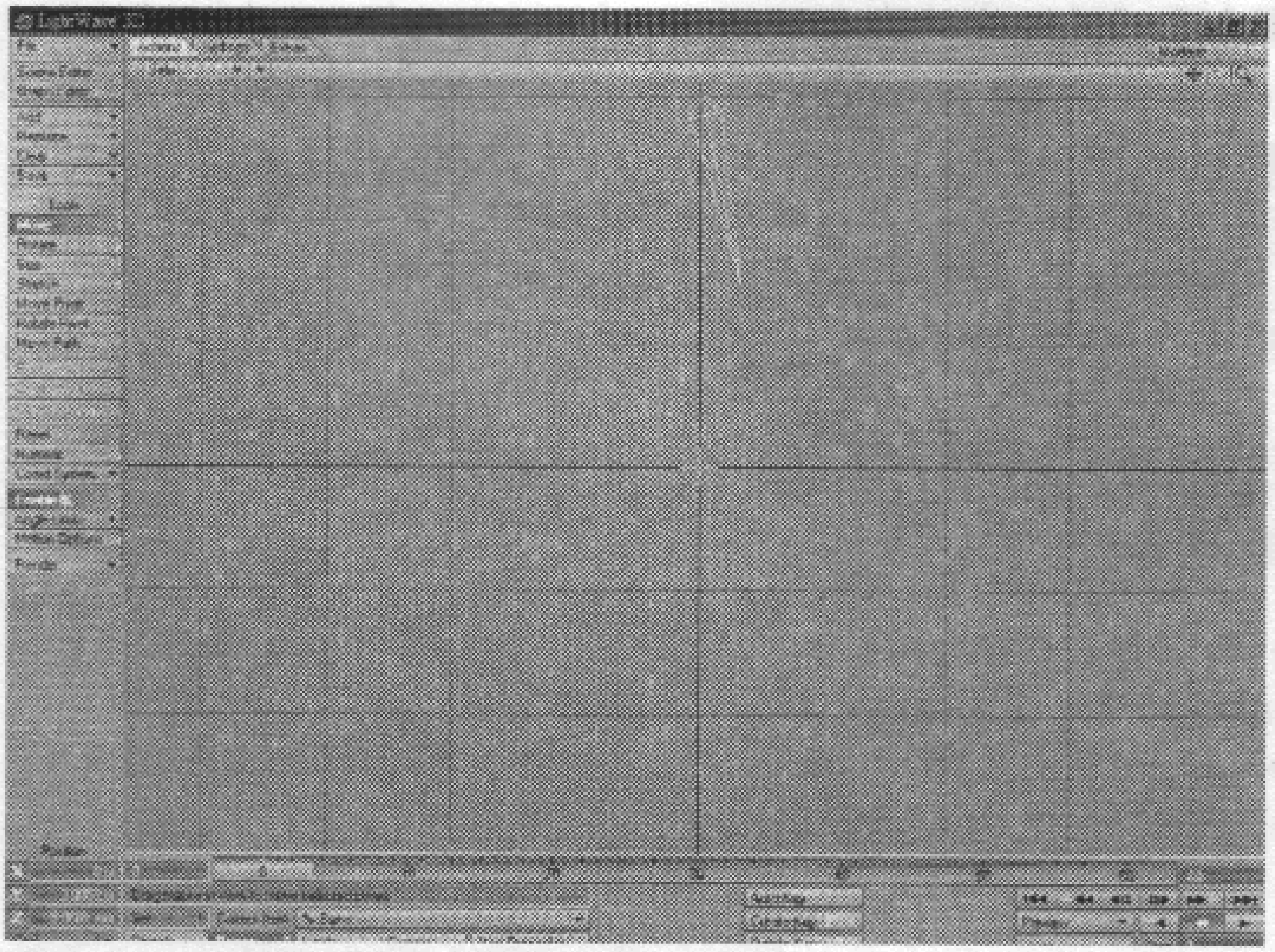
Viewport Layout: Single

3 Hit the 3 key on your keyboard to switch to the Side view.

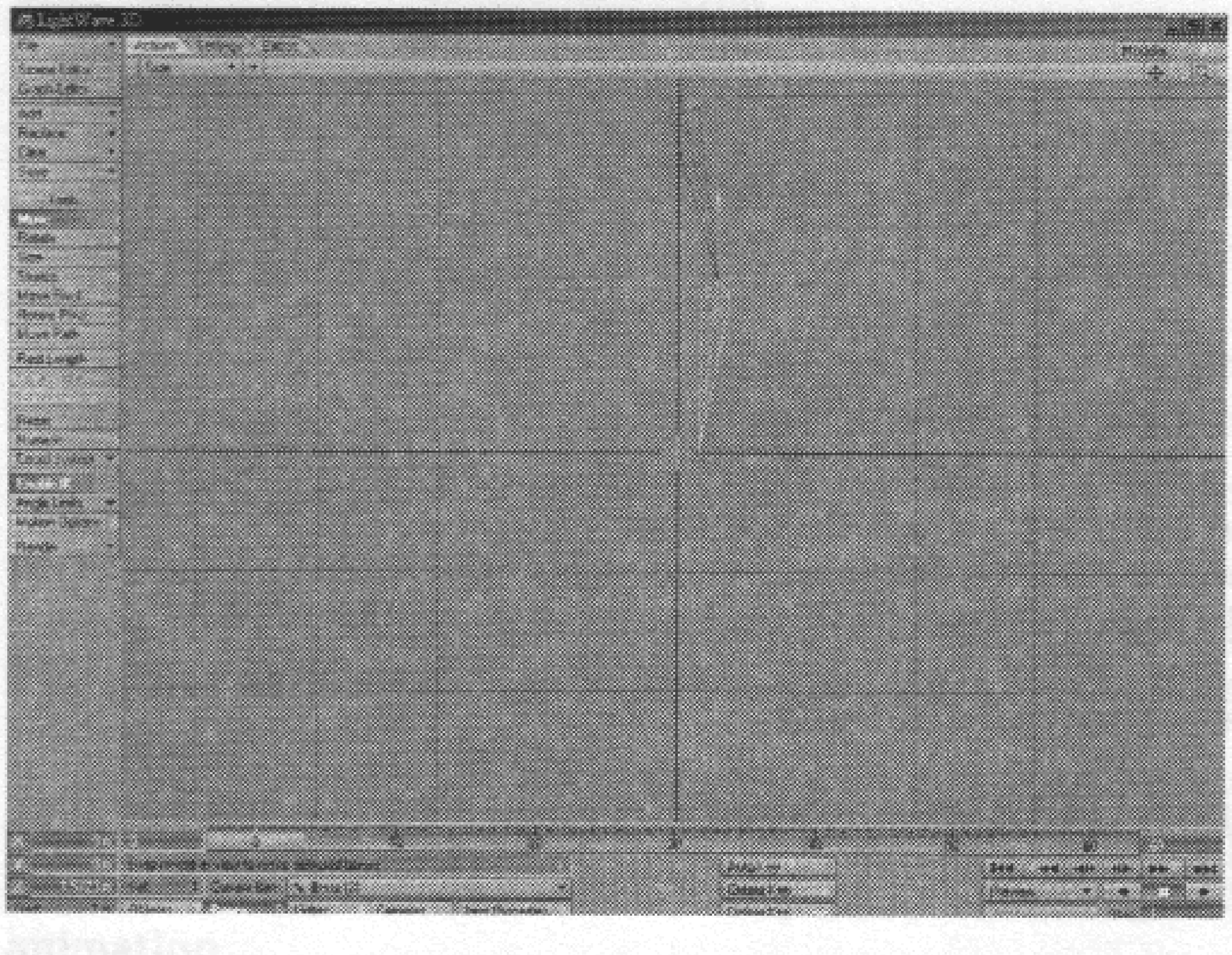
4 Add a null (**Actions** > **Add** > **Add Object** > **Add Null**) and name it BONE DADDY.

5 Select **Actions** > **Add** > **Add Bone** > **Draw Child Bone**.

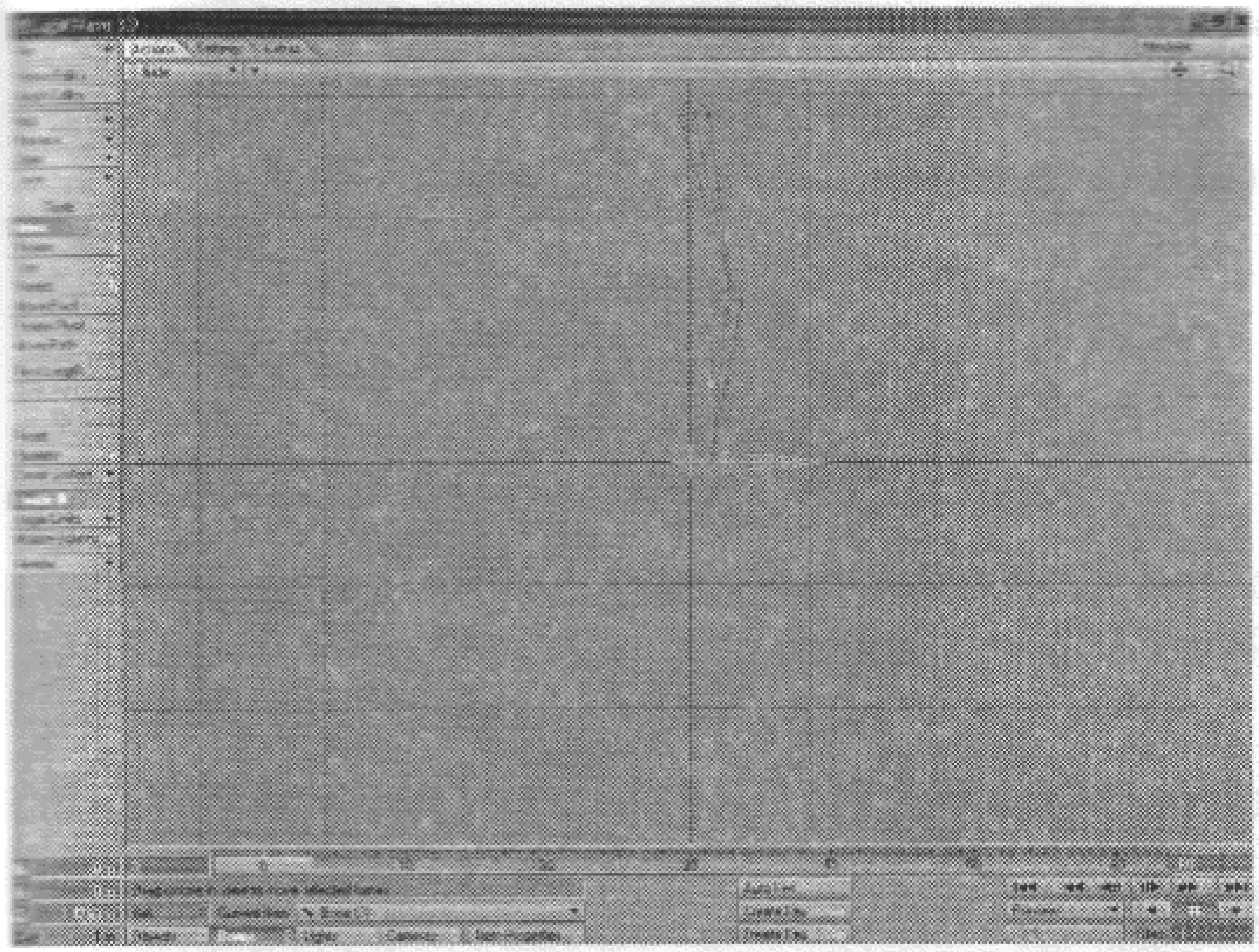
6 Starting near the top of the viewport LMB click and drag down towards the null you just created and slightly to the right. Stop about halfway to the null and release the LMB.



7 Move the cursor down slightly and LMB drag out your second bone. This time drag all the way to 0 on the Y axis just right of the Bone Daddy null.



8 Now move the cursor slightly to the right and LMB drag out the third and last bone in this chain. Try to keep it flat on the Y axis.



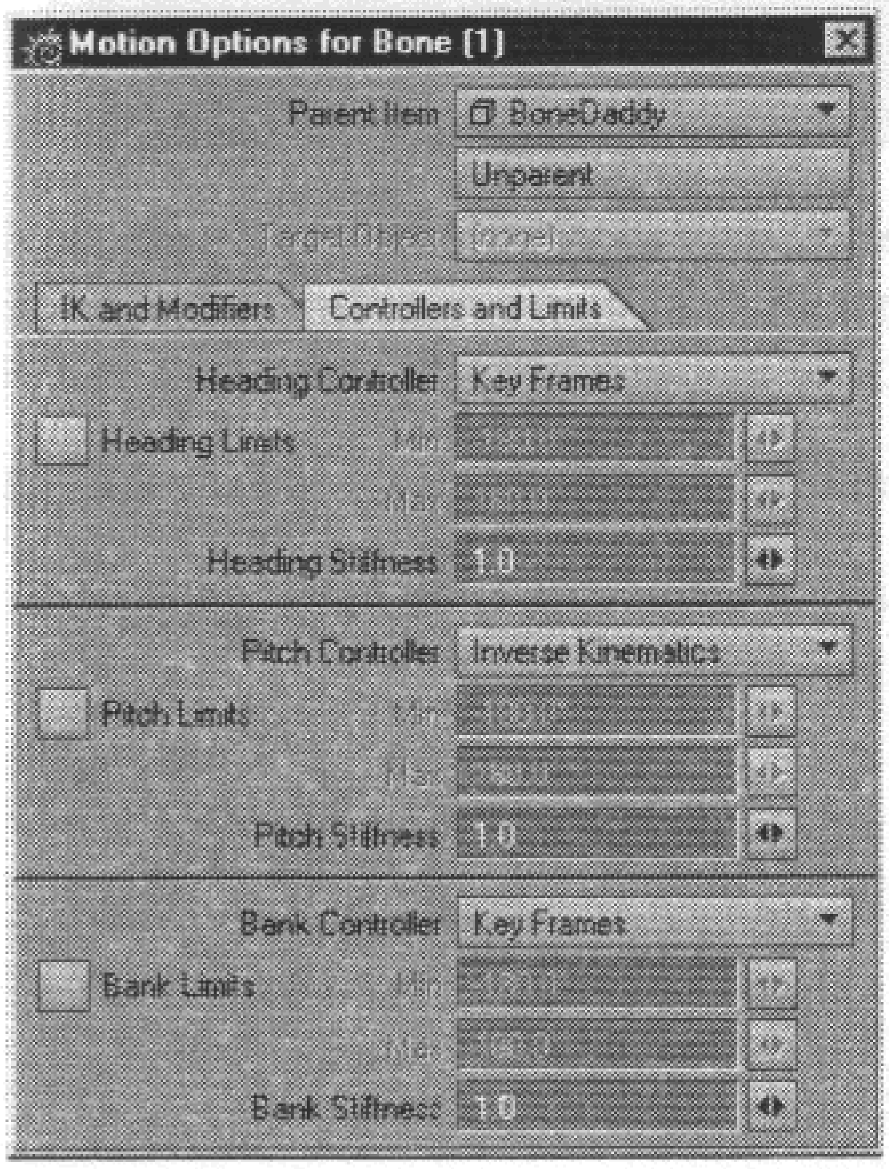
Setting up IK

1 Add a new null and name it GOAL. Leave it at the default location for now.

2 Click on the first Bone in the chain you created earlier.

3 Open the Motion Options panel (**Actions** > **Motion Options** or just hit the M key).

4 Click on the Controllers and Limits tab and select **Inverse Kinematics** from the **Pitch Controller** pop-up menu.



Motion Options for Bone (1)

Parent Item: BoneDaddy

Heading Controller: Key Frames

Heading Stiffness: 1.0

Pitch Controller: Inverse Kinematics

Pitch Stiffness: 1.0

Bank Controller: Key Frames

Bank Stiffness: 1.0

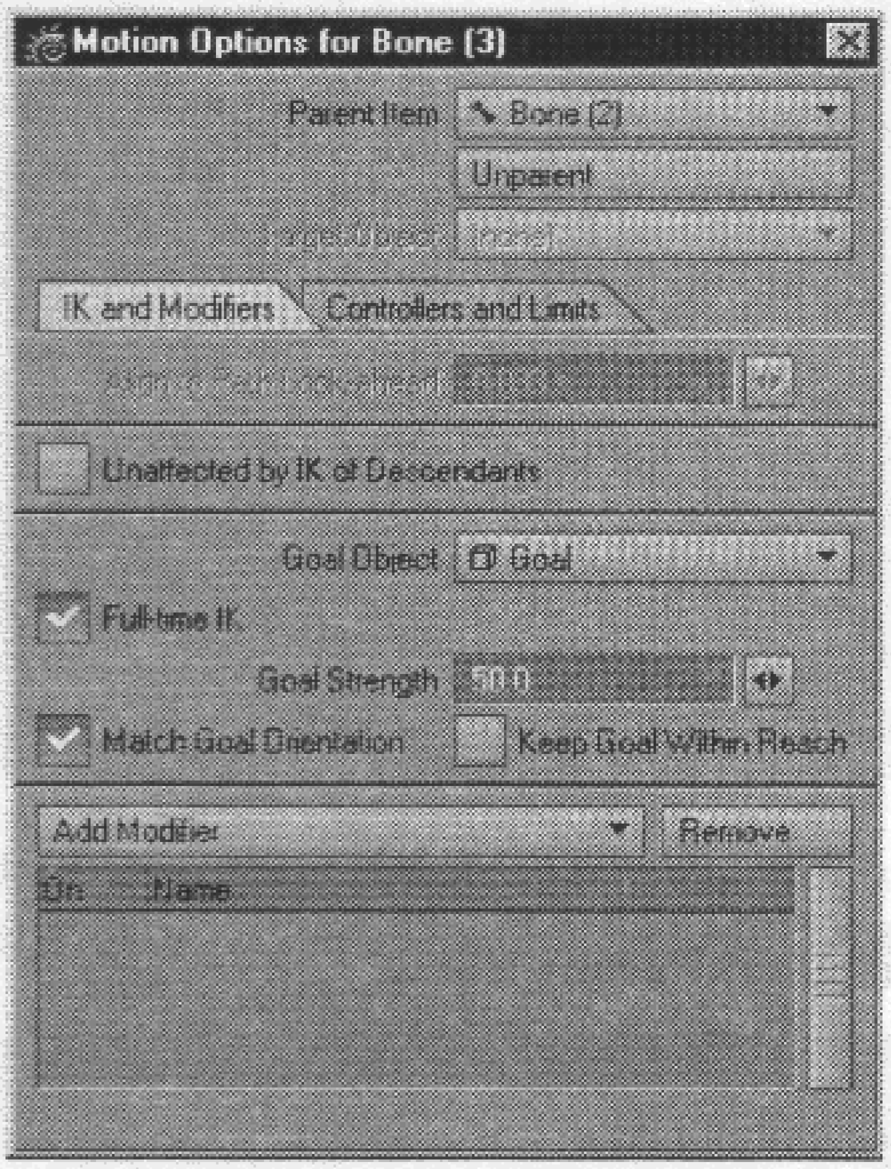
5 In the Layout viewport, select Bone (2).

6 Set Bone (2)'s **Pitch Controller** to **Inverse Kinematics** also.

7 Select Bone (3).

8 Click on the IK and Modifiers tab and set Goal as the **Goal Object** with a **Goal Strength** of 50.

9 Activate **Full-time IK** and **Match Goal Orientation**.



Motion Options for Bone (3)

Parent Item: Bone (2)

Target Object: (none)

□ Unaffected by IK of Descendants

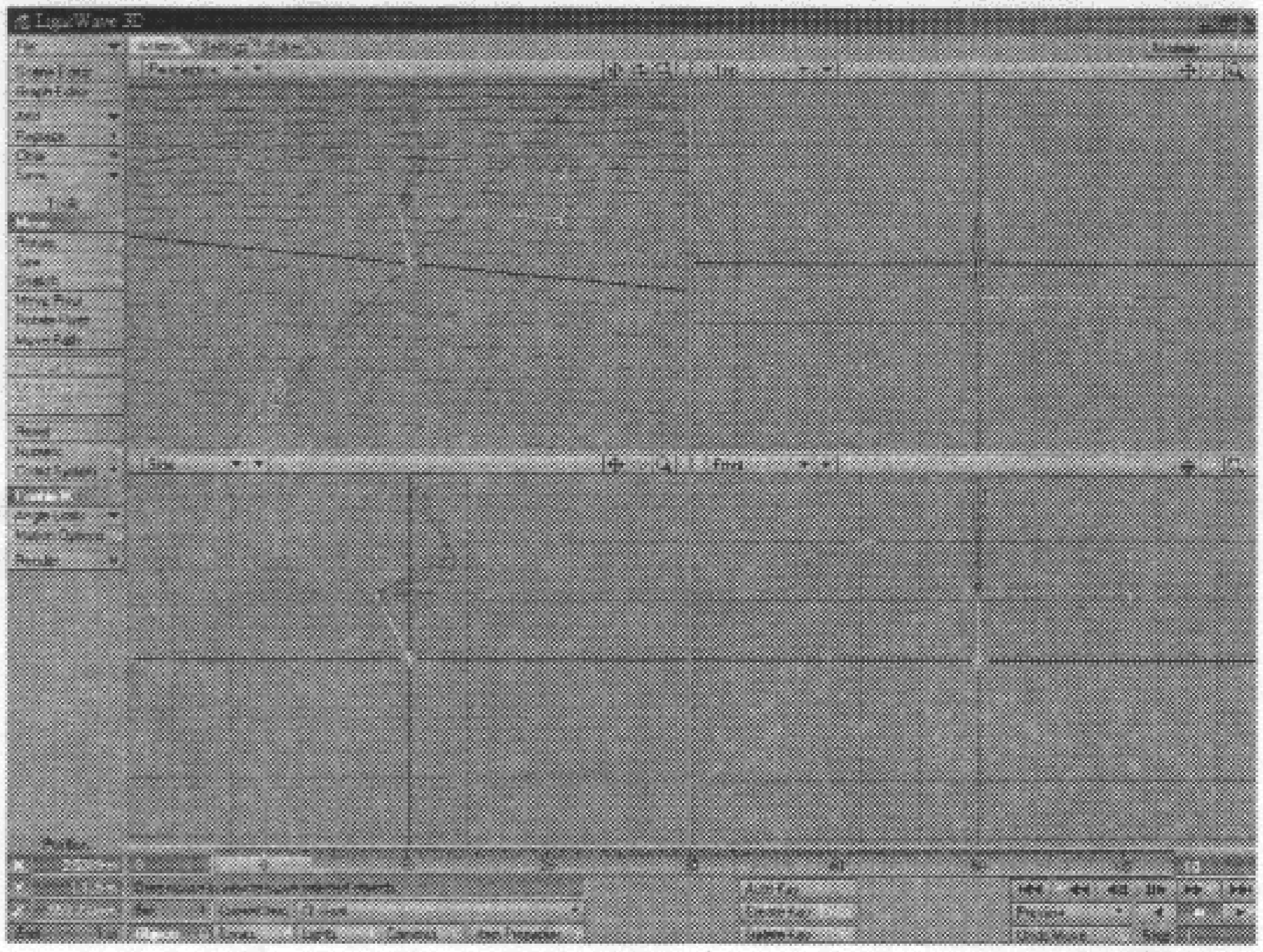
Goal Object: Goal

☑ Full-time IK

Goal Strength: 50.0

☑ Match Goal Orientation □ Keep Goal Within Reach

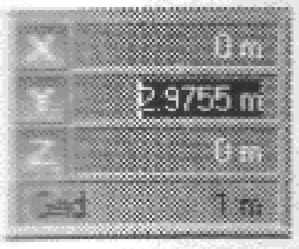
Now we have a working 2D or *Planar IK* chain. If you activate the **Quad Viewport Layout** (SHIFT+F4) and move the Goal, we will notice that the chain cannot reach it if the X value isn't 0. Since we have only activated **Inverse Kinematics** for the **Pitch Controller** of Bone (1) and Bone (2), the chain only moves along the YZ plane. Hence the name *2D IK* (allowing motion along 2 axes) or *Planar IK* (allowing motion along a single plane).



As you've probably already guessed our 2D chain will not be terribly useful in it's current state. We can rectify this situation by giving the entire chain a hinge of sorts. This hinge will allow us to rotate the IK solution plane while maintaining its simplicity.

1 Select Bone (1).

2 Highlight and copy (CTRL+C) its Y value.

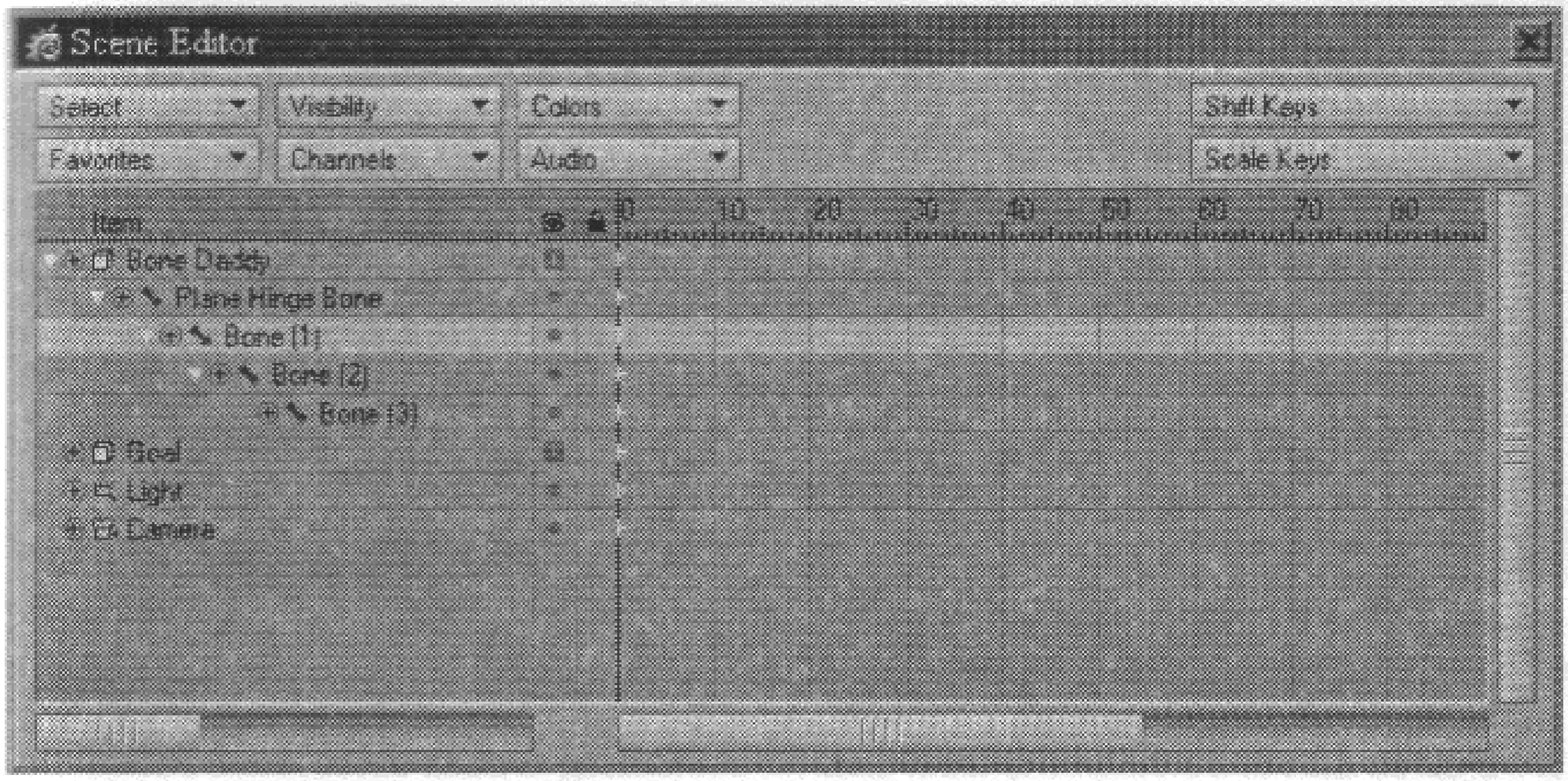


X 0 m
Y 2.9755 m
Z 0 m
Grid 1 m

3 **Add Bone** and **Rename** it PLANE HINGE BONE.

4 Highlight Plane Hinge Bone's Y value and use paste (CTRL+V) to replace the value with Bone (1)'s Y value. Don't forget to **Create Key** (if **Auto Key** is not on). Now Plane Hinge Bone and Bone (1) should be lined up on the Y axis.

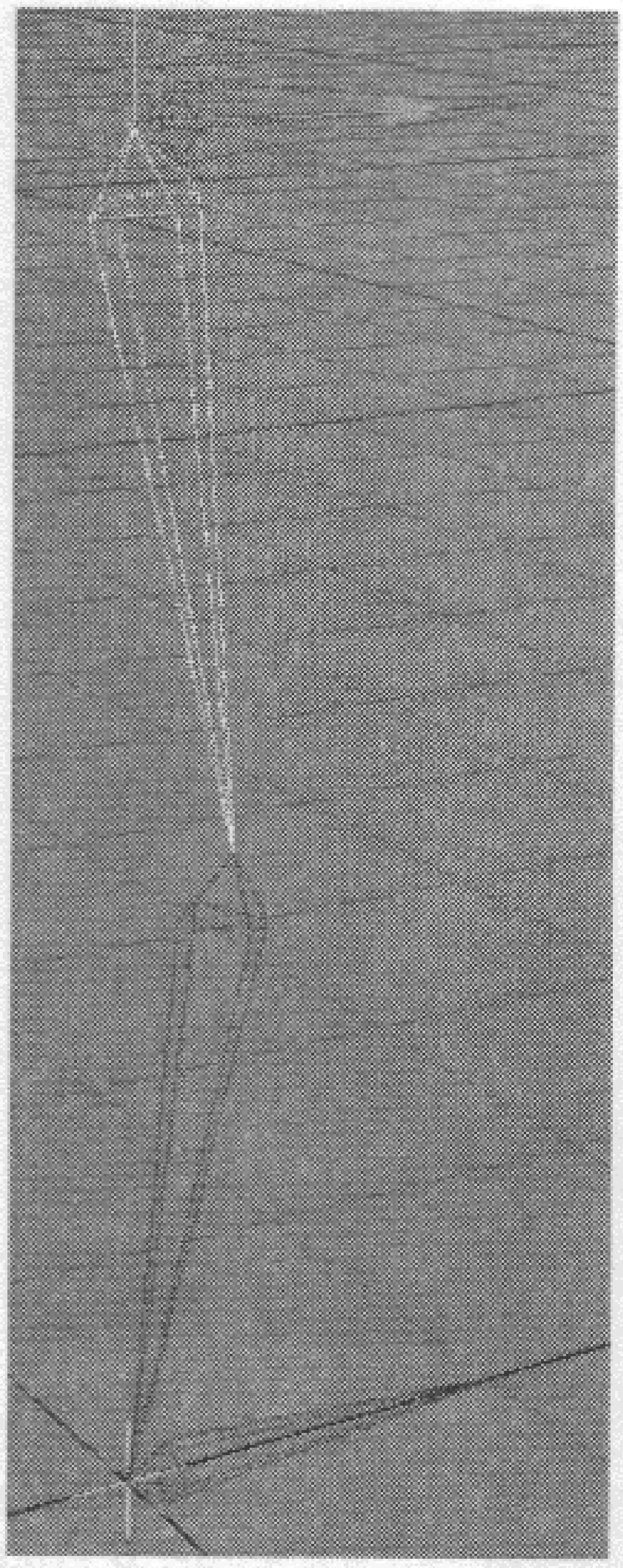
5 In the Scene Editor, Parent Bone (1) to Plane Hinge Bone.



Now Bone (1) seems to have jumped up in the air above Plane Hinge Bone. Don't panic, this is how parenting is supposed to work in Layout. In order to compensate for the *parenting offset* and align the two Bone's Pivot Points, we need to zero out Bone (1)'s Y and Z values.

6 Select Bone (1) and reset its Y and Z values to 0.

7 **Create Key**. Your Bone chain should now look something like this.



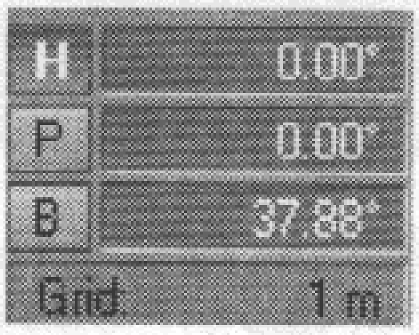
8 Select Plane Hinge Bone and open the Motion Options panel.

9 Set the **Bank Controller** to **Inverse Kinematics**. Leave **Heading Controller** and **Pitch Controller** on **Key Frames**. Now our chain is free to follow the Goal as it moves along the X axis. There is another advantage to adding the Plane Hinge Bone as the root of our bone hierarchy. By rotating it on heading, we can reorient the entire chain and its IK solution plane. This gives us direct control over which direction the knee is pointing. **Match Goal Orientation** keeps the foot Bone aligned with its Goal no matter how the leg chain is rotated.

The Finishing Touches

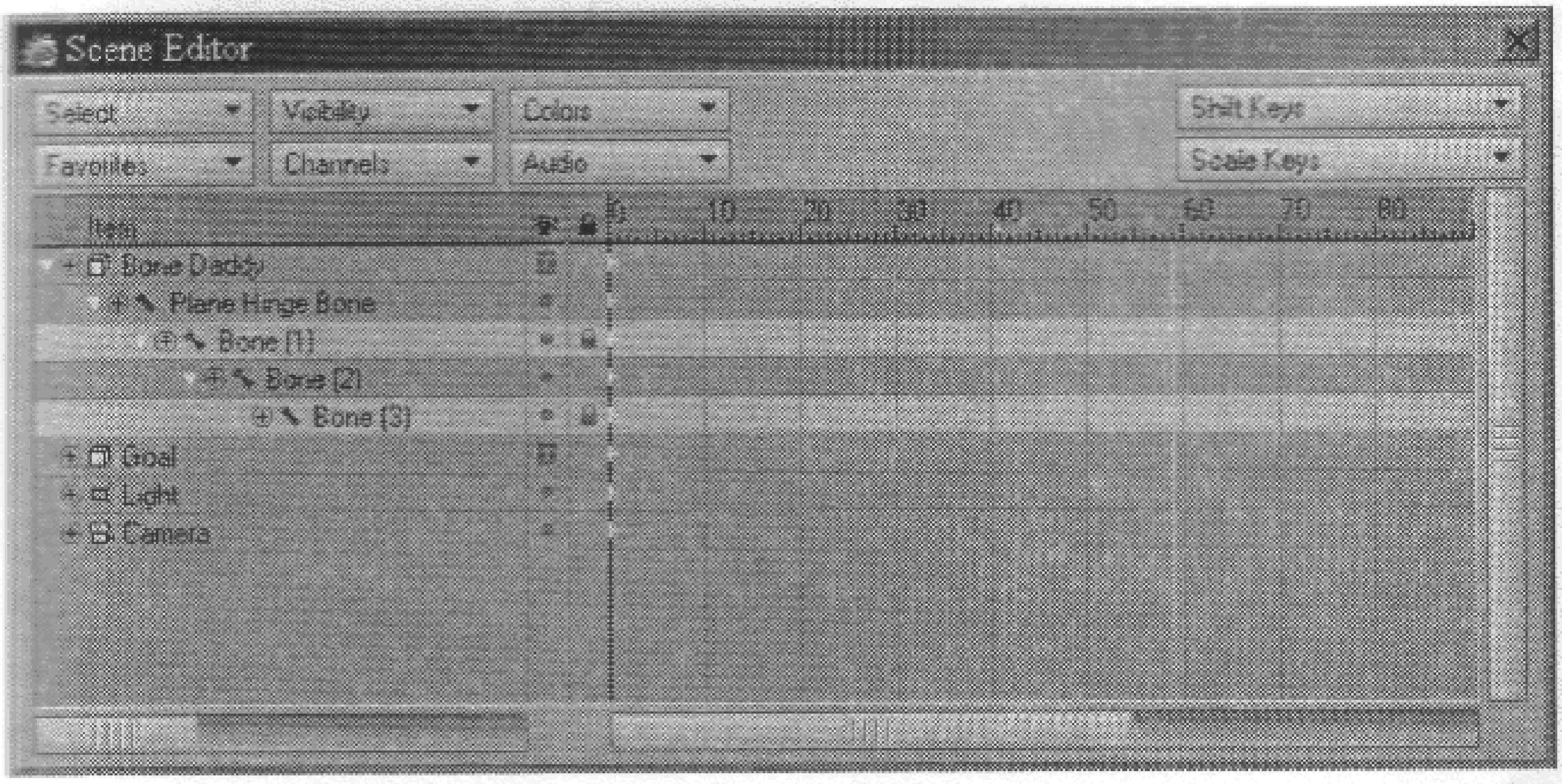
Now we have a planar IK chain capable of tracking a Goal along all three axes. Still, there are a few things we can do to make using it a bit easier.

1. Select Plane Hinge Bone and turn off its Pitch and Bank channels. Bank is being controlled by IK and we don't want to accidentally rotate Pitch.



H	0.00°
P	0.00°
B	37.88°
Grid	1 m

2 Set the selection locks for Bone (1) and Bone (3) in the Scene Editor. This will keep them from being selected accidentally with the mouse.



Now you are ready to play. Experiment with posing the leg. You should find it easy to control the leg with just the Plane Hinge Bone and the Goal. Notice that we did not set any rotation limits for any of the Bones. Because we drew the leg bones—which are the only bones in the chain with IK rotation controllers—slightly bent, the IK solution will attempt to bend in the same direction. You may find during your experimentation that the IK yields desirable results without benefit of **IK Limit** or **Stiffness** settings.

Planar IK Leg Setup

This tutorial focuses on setup issues directly related to the movement of bones in a hierarchy. There is a separate tutorial focusing on issues directly relating to how bones deform geometry.

Loading Reference

1 Open the Backdrop panel (**Display** > **Backdrop**)

2 Load L6_QUINCY_BKG_PLACEMENT_BOX.LWO from the TUTORIALS\QUINCY directory. We will use this to aid us in sizing the backdrop images. Then use the Image Editor to load QUINCY_TOP_REF.TGA, QUINCY_FRONT_REF.TGA and QUINCY_SIDE_REF.TGA from the TUTORIALS\QUINCY folder.

3 In the Backdrop panel, set Viewport 1's **Image** to QUINCY_TOP_REF.TGA and click **Automatic Size**. Change the first **Size** field from 930.33 mm to -930.33 mm, this inverts the image on the X axis so it displays properly in the Back viewport.

4 Set Viewport 3's **Image** to QUINCY_FRONT_REF.TGA and click **Automatic Size**. Change the first **Size** field from 930.33 mm to -930.33 mm, this inverts the image on the X axis so it displays properly in the Back viewport.

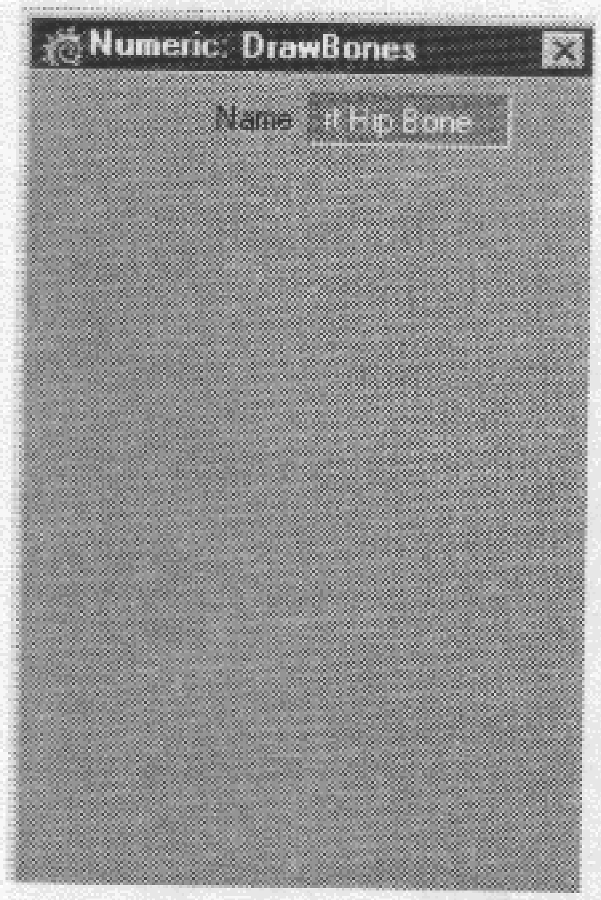
5 Set Viewport 4's **Image** to QUINCY_SIDE_REF.TGA and click **Automatic Size**. Now you can use **Objects** > **File** > **Close Object** to get rid of the placement box.

6 You should now have three images appearing in the Top, Back and Right viewports. This will be our reference.

Drawing Skelegons

1 Select **Additional** > **DrawSkelegons**.

2 In the DrawSkelegons **Name** field on the Numeric panel type RT HIP BONE.



Numeric: DrawBones

Name: rt Hip Bone

3 Starting in the Right viewport at the top of Quincy's leg, hold the CTRL key and LMB click and drag to the right. As you drag a Skelegon will appear. Holding the CTRL will constrain our motion to the Z-axis.

4 Hit the SPACEBAR to drop the **DrawSkelegons** tool.

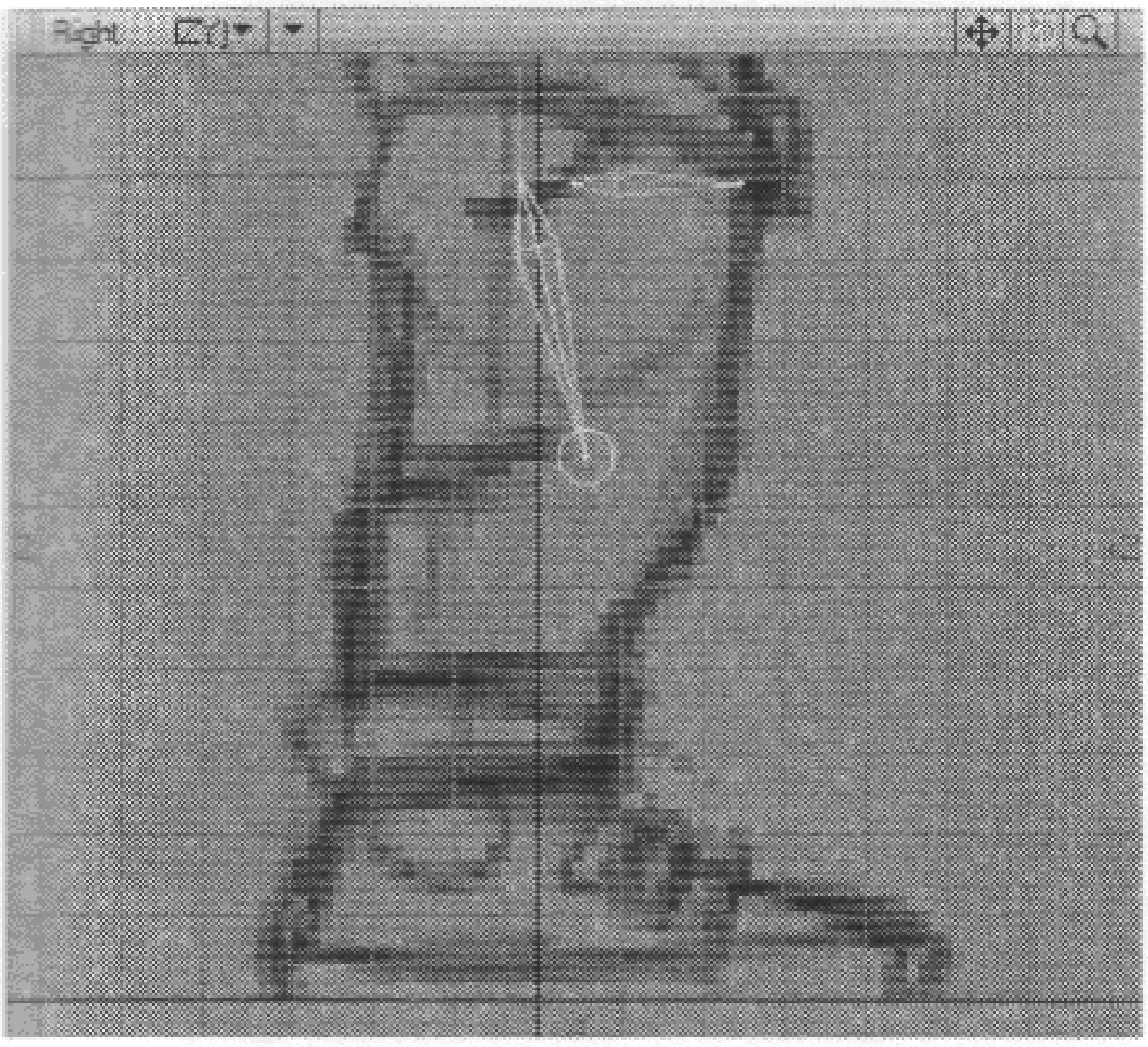
5 Now re-select **DrawSkelegons**. This will allow us to start a new Skelegon chain with a new name.

6 Adjust the Right Viewport so Quincy's leg fills the the window.

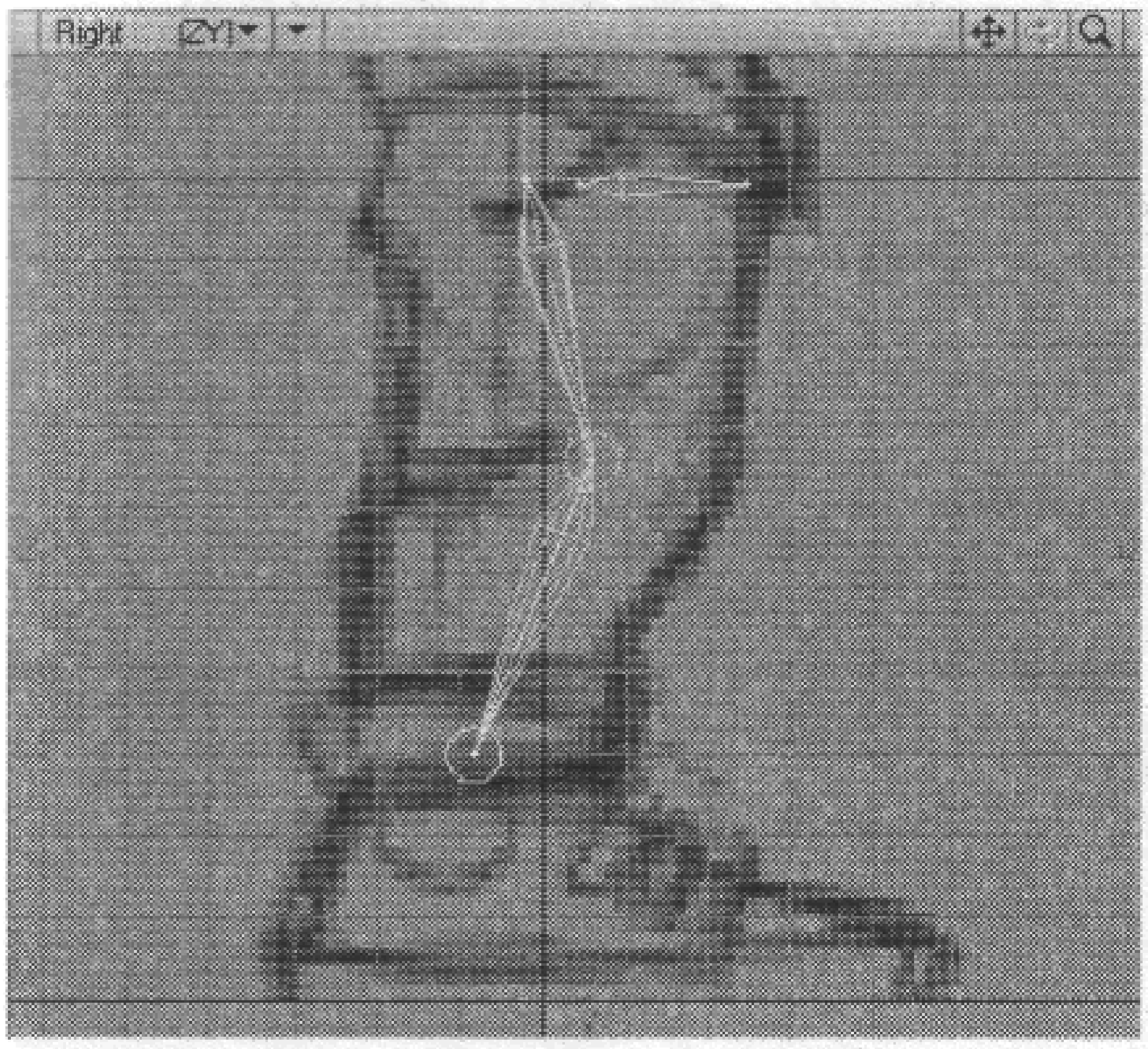
7 The **DrawSkelegons** tool should still be selected. Type RT LEG BONE in the DrawSkelegons **Name** requester on the Numeric panel.

8 In the Right viewport LMB click to the left of the first point in the rt Hip Bone you just created and drag down toward the knee in the Quincy reference drawing.

9 Release the mouse button when you reach the knee, but don't drop the DrawSkelegons tool.

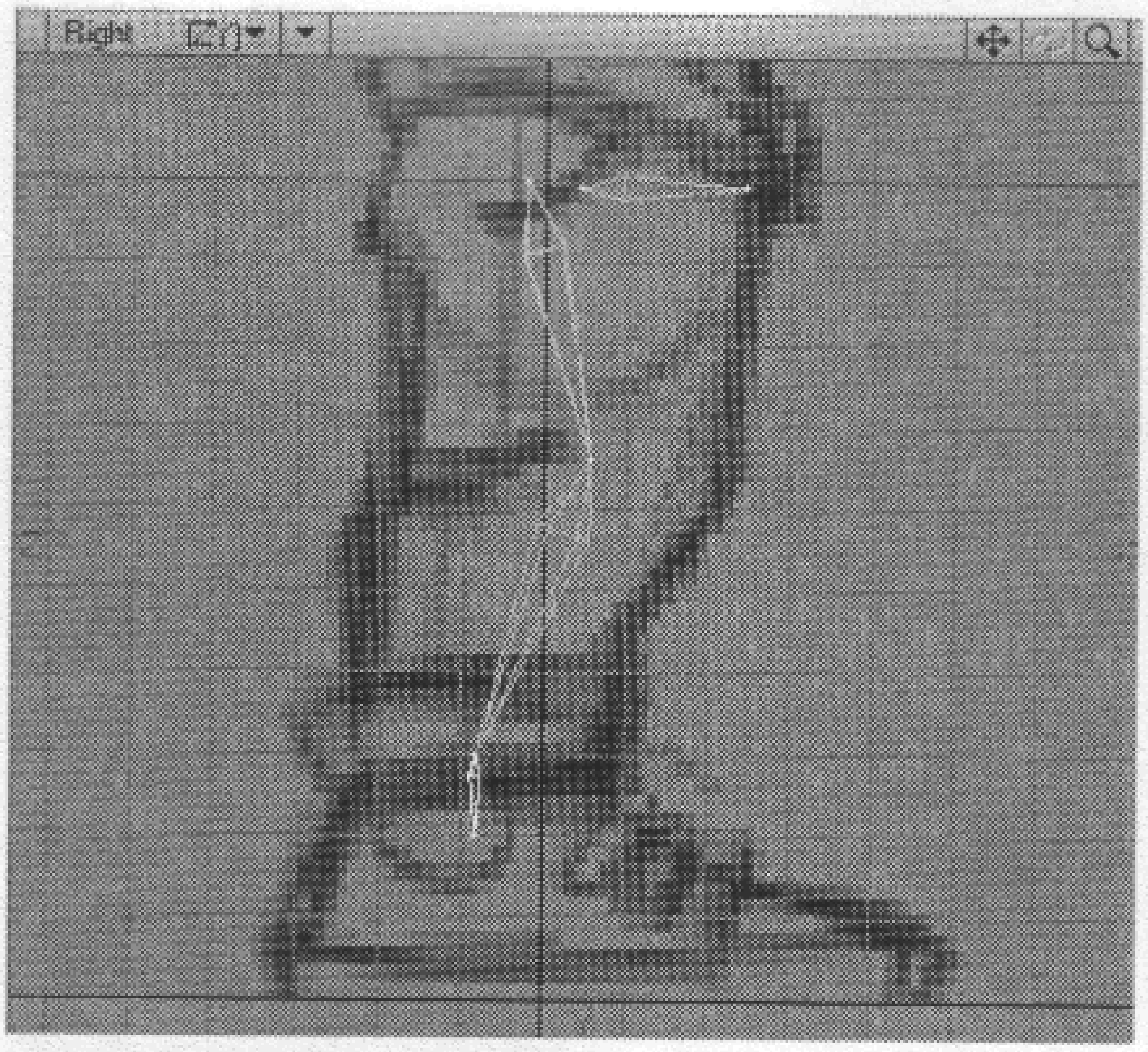


10 Repeat this process. This time drag down to Quincy's ankle.



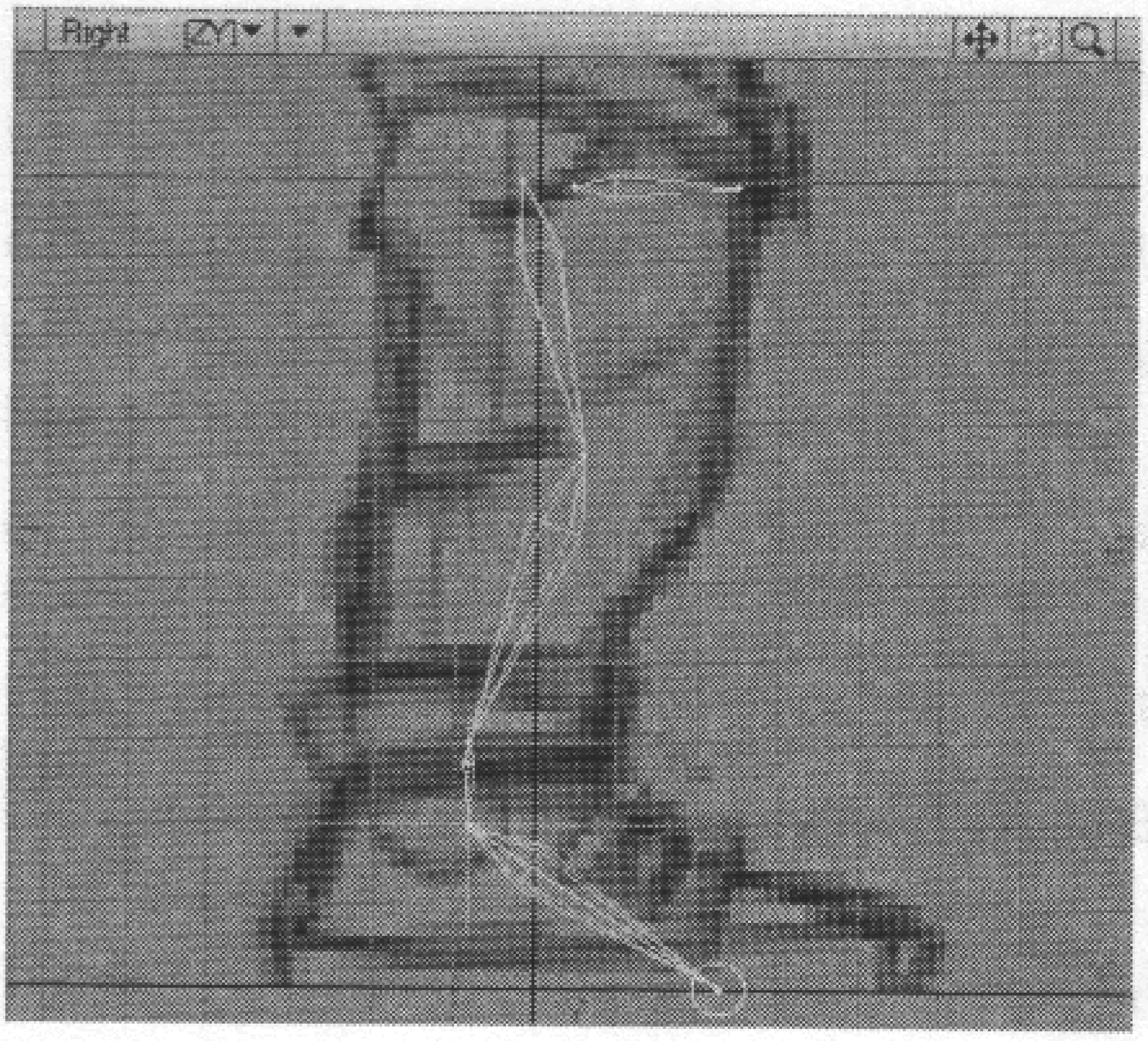
11 Now, while holding down the CTRL key, LMB click directly beneath the point you just created and drag straight down.

12 Drop the **DrawSkelegons** tool and select the Ankle Skelegon you just drew.

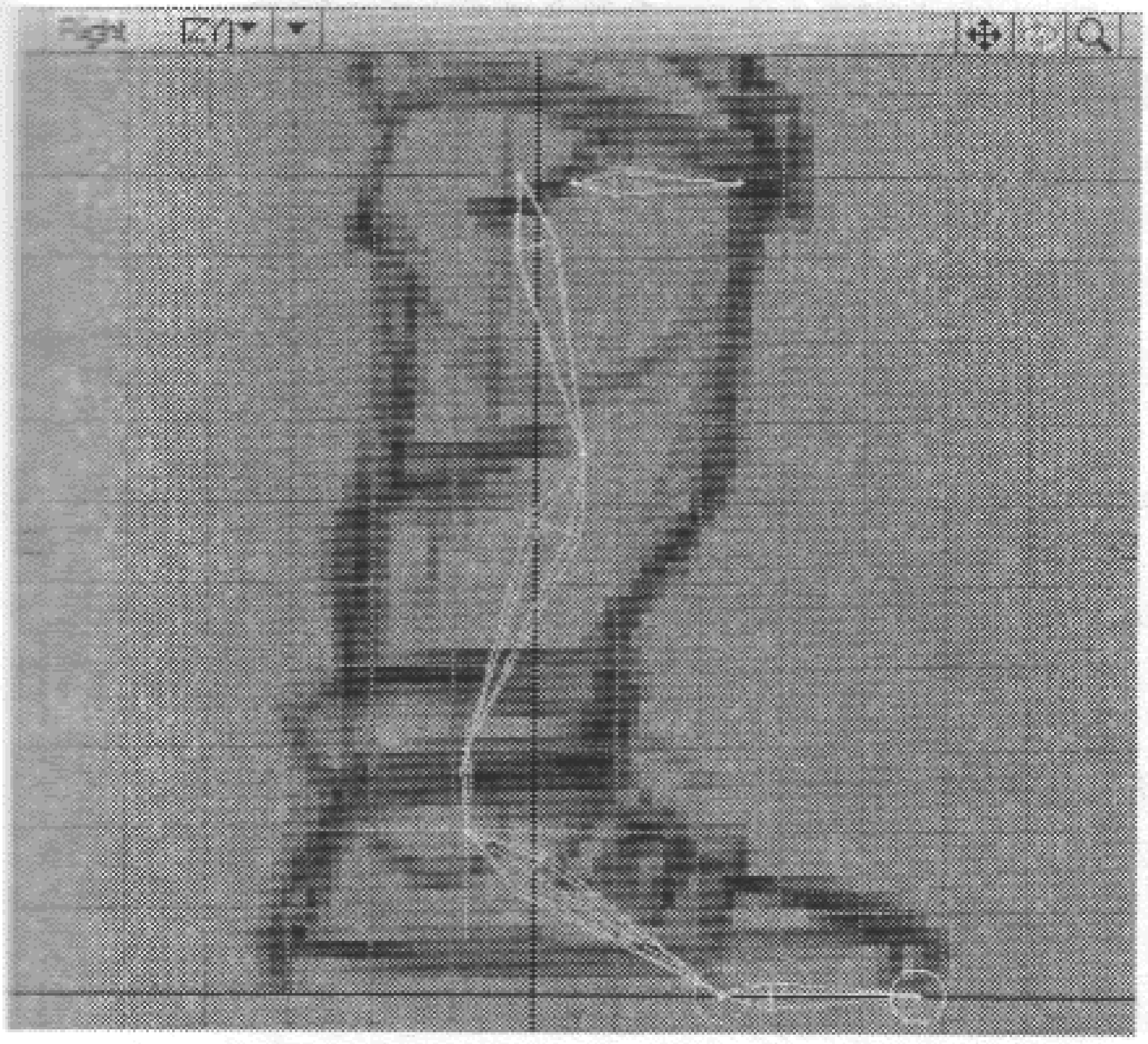


13 Re-activate the **DrawSkelegons** tool and type RT TOE BONE in the Numeric panel.

14 In the Right viewport, LMB click and drag down and to the right. You will notice that the new Skelegon is automatically connected to the Ankle Bone we selected. Drag all the way to the ground (0 on the Y axis) just below the base of the toe. In reality, this joint belongs higher on the Y but placing it on the floor makes set-up easier.

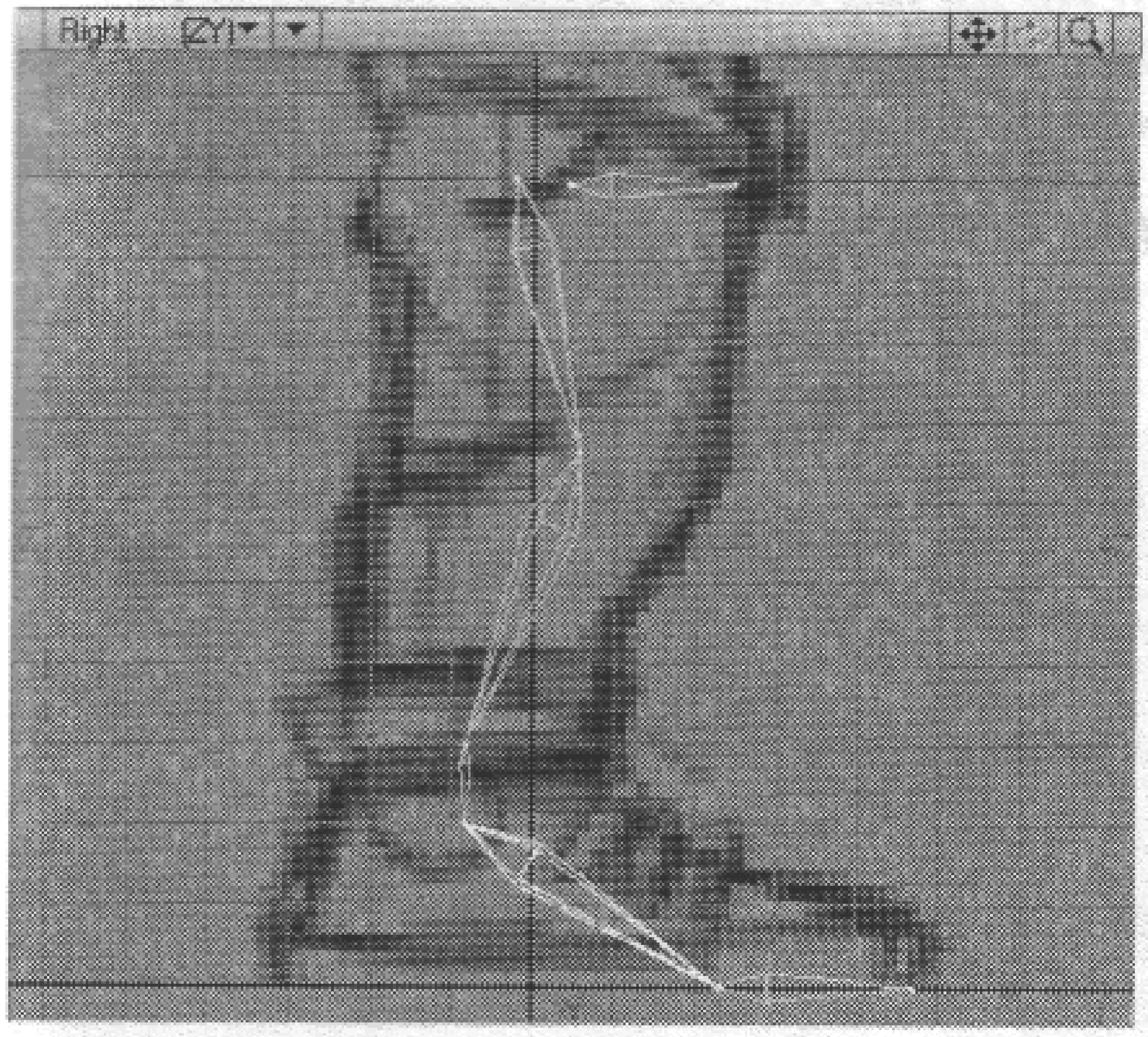


15 While holding CTRL, LMB click and drag to the right to create a Skelegon along the floor. End the Skelegon right at the tip of Quincy's toe in the reference drawing. This will be important later for animation setup.



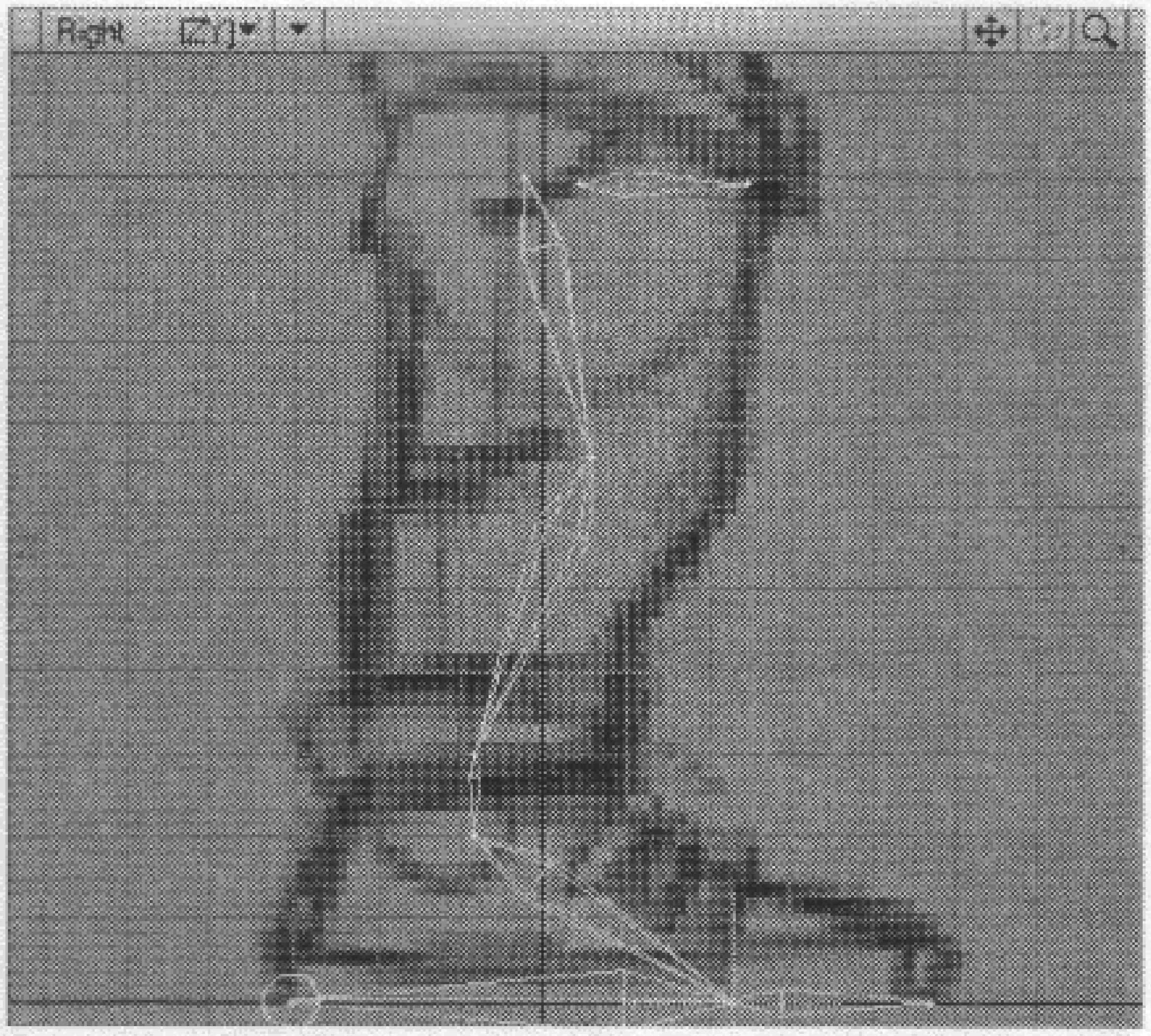
16 Hit the SPACEBAR to drop the **DrawSkelegons** tool.

17 Select the diagonal Skelegon you just drew between the ankle and the floor.



18 Activate **DrawSkelegons** and CTRL+LMB to the left—don't release the mouse button until the Skelegon reaches the back of Quincy's heel in the drawing.

19 Before you drop the **DrawSkelegons** tool. Type RT HEEL BONE in the DrawSkelegons **Name** requester on the Numeric panel. This will rename the current chain even though you have already drawn both heel bones. **DrawSkelegons** will always apply the current name to all bones in the current chain being created. The name in the **DrawSkelegons** Numeric requester will not be applied to previously drawn Skelegons. That is why I told you to drop the tool each time we created a new chain.



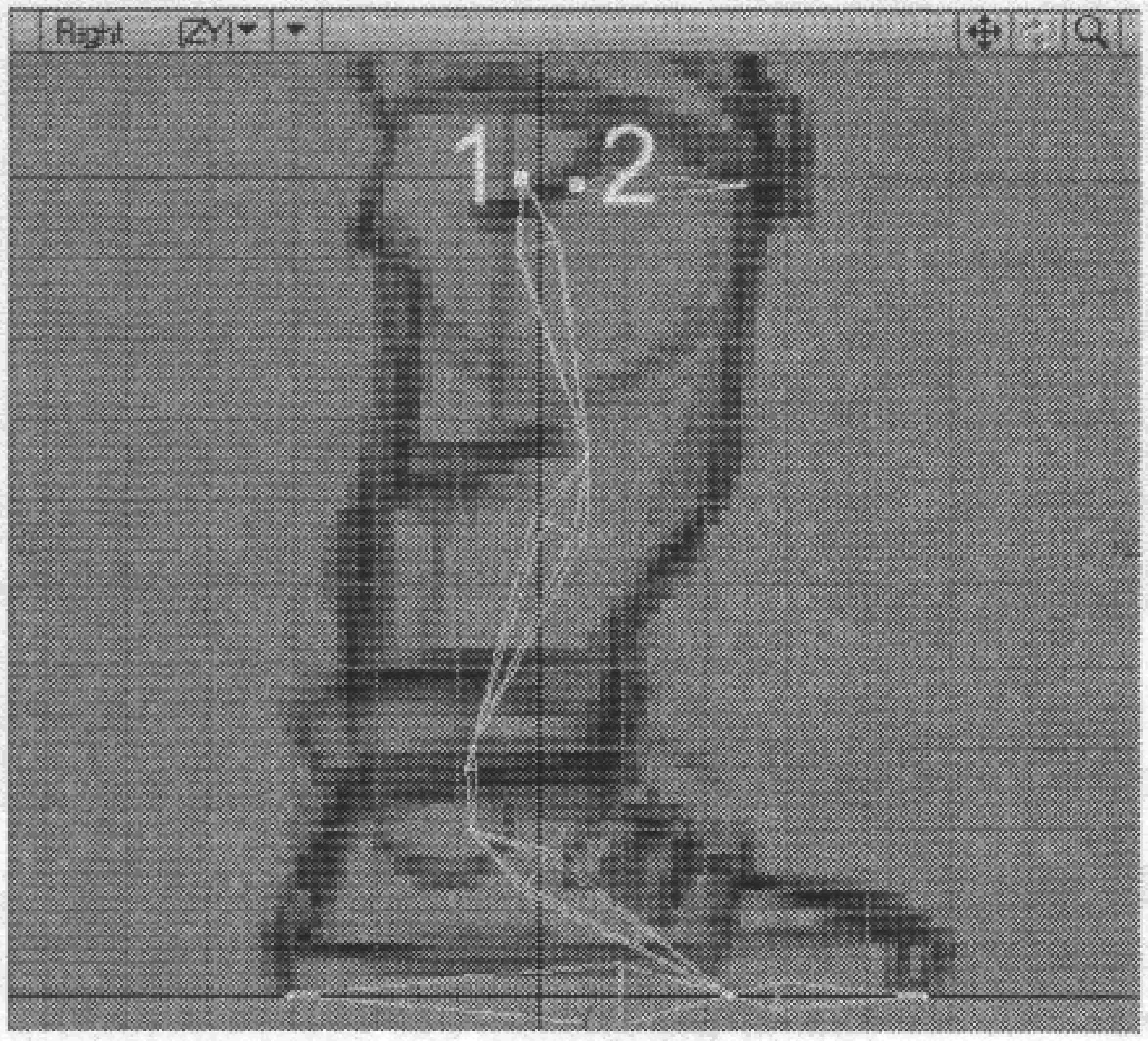
Tidying up the Skelegon Hierarchy

Now you have all of the Skelegons you will need to create your leg hierarchy. However, there are still a few things we need to do before heading to Layout though.

First things first. We want the pivot point for the first Skelegon in the rt Leg Bone chain to match the one in the rt Hip Bone exactly. We can do this by welding the points together.

1 Select the starting point of rt Leg Bone.00 (the first bone in the Leg chain).

2 Select the first point in rt Hip Bone.00 (the only hip bone).



3 Select **Tools** > **Points** > **Weld** (Ctrl+W) to connect the points. Selection order is important. If you had selected the end point of rt Hip Bone.00 first, it would have been snapped to the first point in the rt Leg Bone.00.

The next thing we want to do is rename a couple of bones.

1. Select **Objects** > **Additional** > **SkelegonTree**. A panel pops up showing a list of the Skelegons you just created. Note the indentation. It indicates the hierarchy. Each bone is a child of the bone listed before it with one exception. rt Leg Bone.00 is not a child of rt Hip Bone.00. This is because we dropped the **DrawSkelegons** tool after drawing rt Hip Bone.00 and did not select rt Hip Bone.00 before drawing the rt Leg Bone chain.

2. Double-click on rt Leg Bone.02. This opens the Rename Skelegon requester.

3. Replace the name RT LEG BONE.02 with RT ANKLE BONE. It isn't necessary to add a number.

4. Click **OK** to close the requester. Now you will see rt Ankle Bone in the SkelegonTree panel in the place of rt Leg Bone.02.

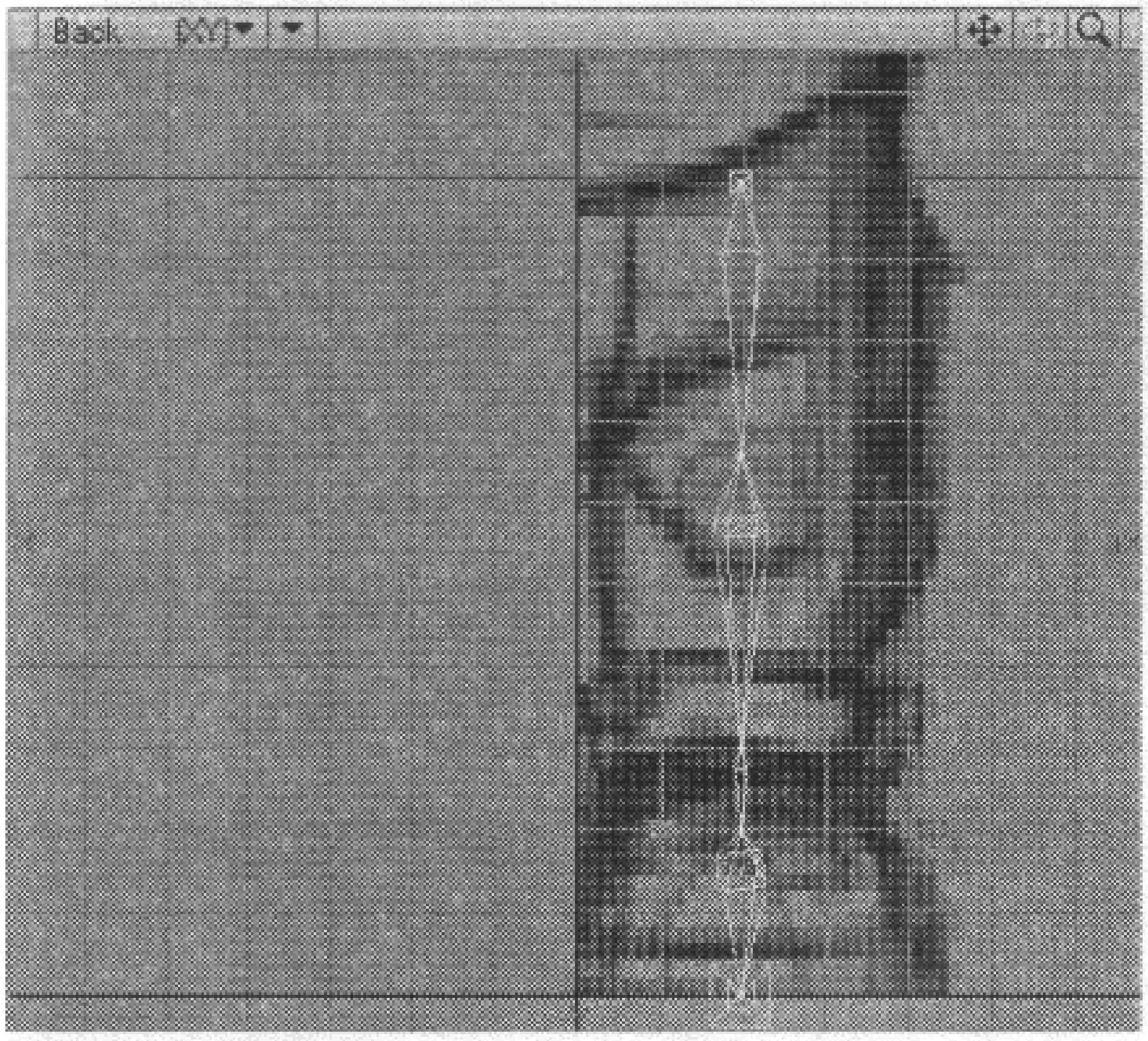
5. Repeat the renaming process with RT TOE BONE.00 changing it to RT FOOT BONE.

6. Close the SkelegonTree panel.

You may have noticed that the all of the Skelegons we have created are *smack dab* in the middle of the character in the Back and Top views (if the Back and Top Viewports were not centered on the X axis while you were creating your Skelegons, some of them will have different X values.) Let's take care of that right now. Hit T to select the **Move** tool.

1. If all of your Skelegons are not flat on the X axis use **Tools** > **Points** > **Set Value** to set them all to 0 on the X axis.

2. While holding CTRL, LMB click anywhere in the Back viewport (it is not necessary to click directly on the Skelegons) and drag to the right until the Skelegons align with the center of Quincy's right leg.



Connecting a Pelvis

If we were only building a single leg, what we have now would be fine but we're going to build a complete character with two legs, two arms and a spine. The root of the entire hierarchy will be the pelvis.

1 Make sure nothing is selected and copy (C) the contents of the current Layer (your entire Skelegon leg).

2 Choose **Objects** > **File** > **Close Object** to get rid of your original leg object. Don't worry we've got it copied in memory.

3 Load L6_QUINCY_PELVIS_SKELEGONS.LWO.

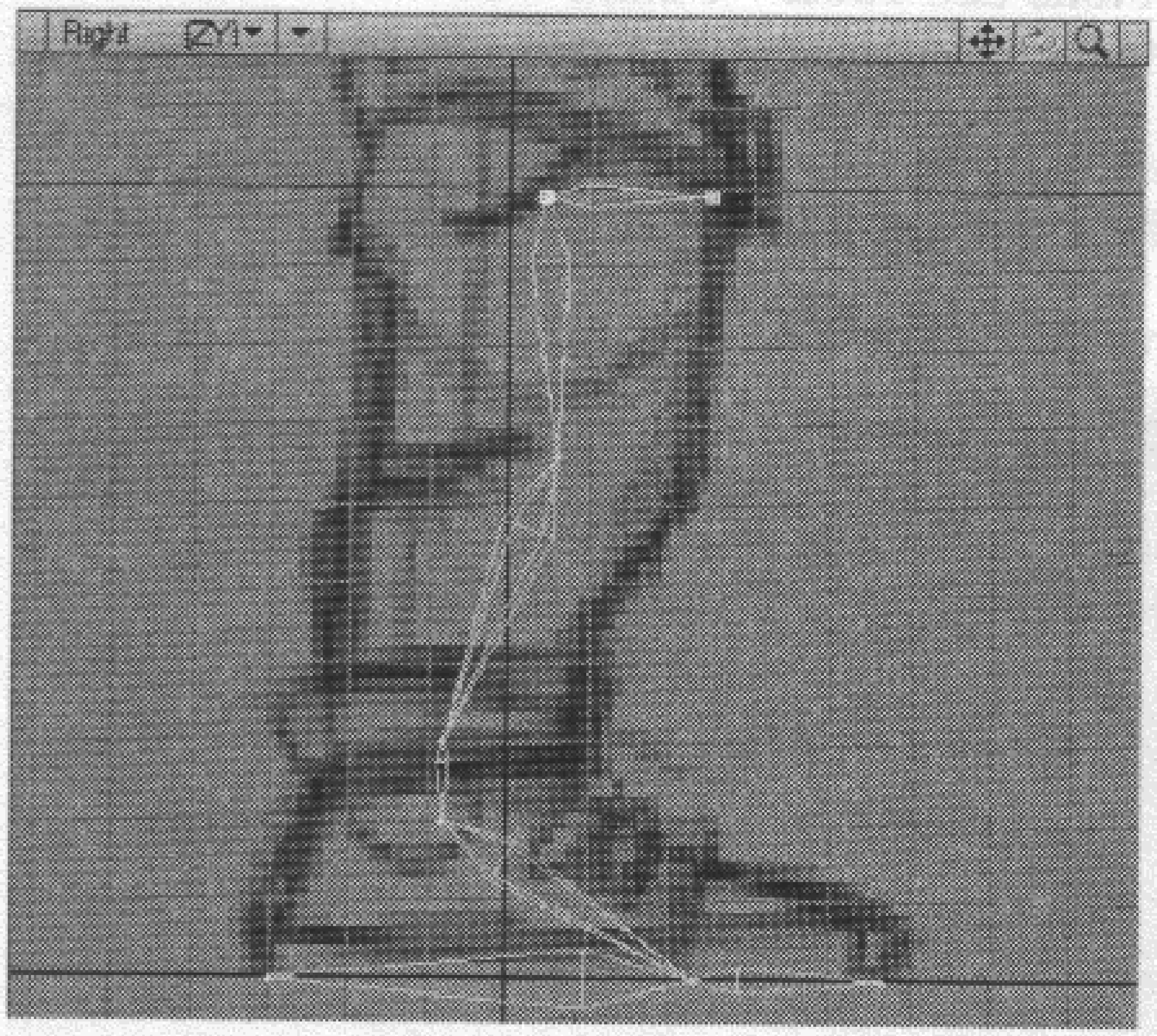
4 Paste your leg Skelegons into Layer 1 of L6_Quincy_pelvis_Skelegons.lwo with the three pelvis Skelegons.We need to connect our leg Skelegons to the end of rt Hip Connector Bone.01. We also want to keep the rt Hip Connector chain and the rt Leg chain flat on their respective planes in order to achieve the desired bone orientation in Layout.

5 Select any Skelegon in the original leg Skelegon hierarchy and click **Tools** > **Selection** > **Sel Conn** (]) to select all connected Skelegons.

6 Hit the = key (**Display** > **Visibility** > **Hide Unsel**) to hide the Pelvis and rt Hip Connector Skelegons.

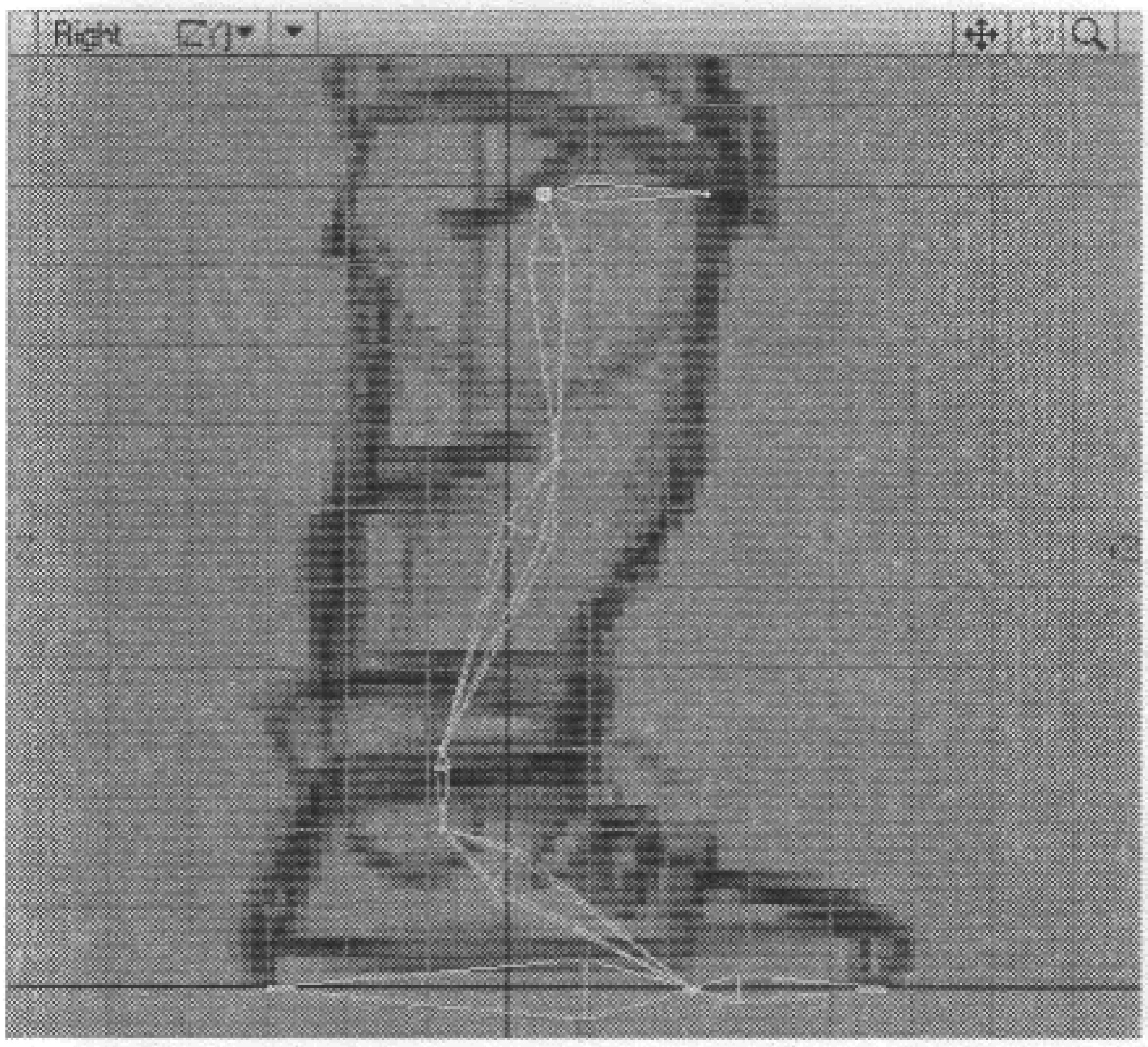
7 Use **Set Value** to place your entire leg chain at 100 mm on the X axis.

8 Switch to Point Selection mode (CTRL+G) and select the two points in the rt Hip Bone.



9 **Set Value** to 495.3 mm on the Y axis.

10 Deselect the end point and **Set Value** to 22.7 mm on the Z axis.



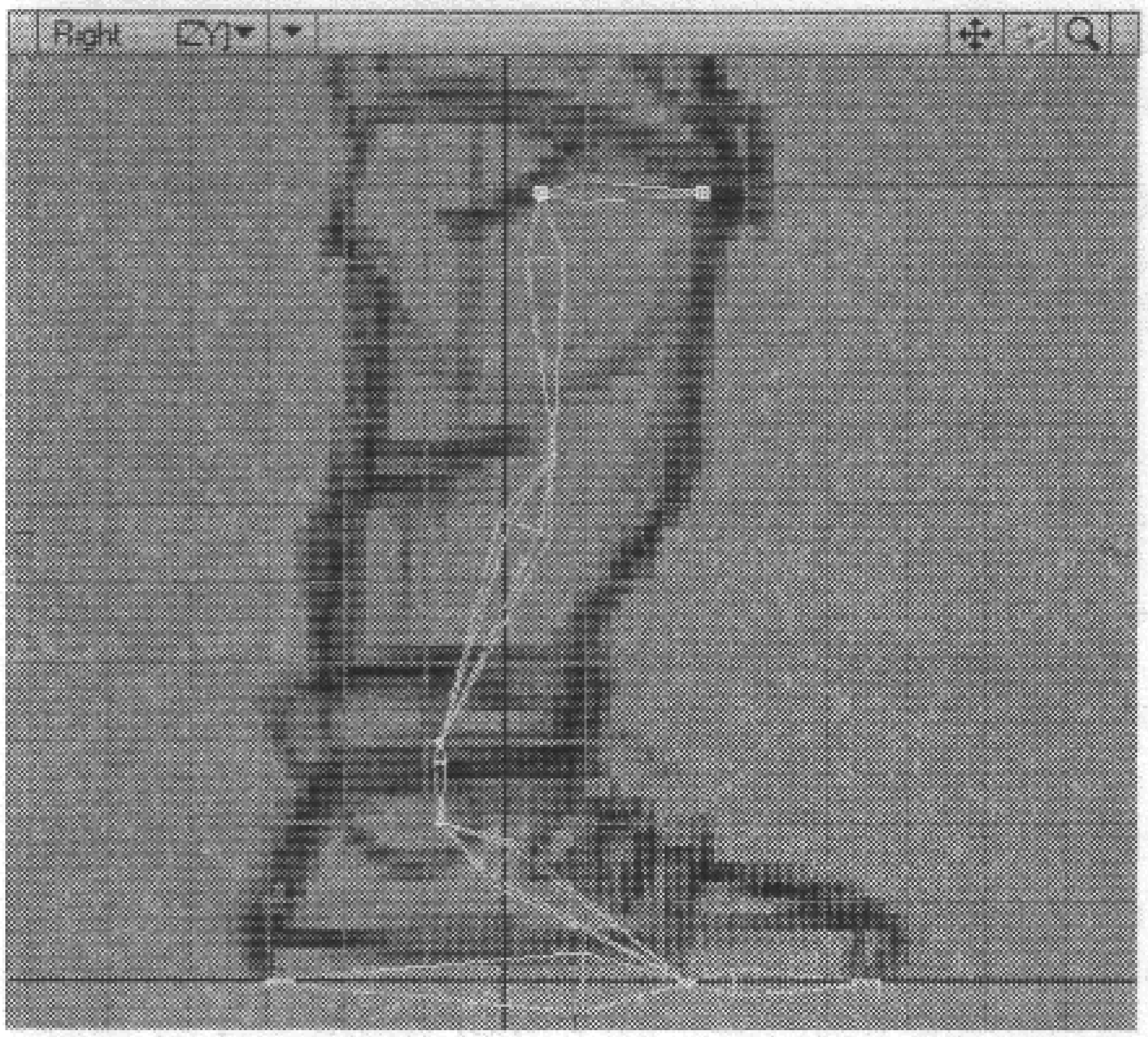
11 Deselect all points and hit the \ key (**Display** > **Unhide**). Now the base of the hip bone is perfectly aligned with the end of rt Hip Connector Bone.01.

12 Select **Tools** > **Points Merge** (M). Leave **Type** set to **Automatic** and click **OK**. One point should have been eliminated.

6 Hit the = key (**Display** > **Visibility** > **Hide Unsel**) to hide the Pelvis and rt Hip Connector Skelegons.

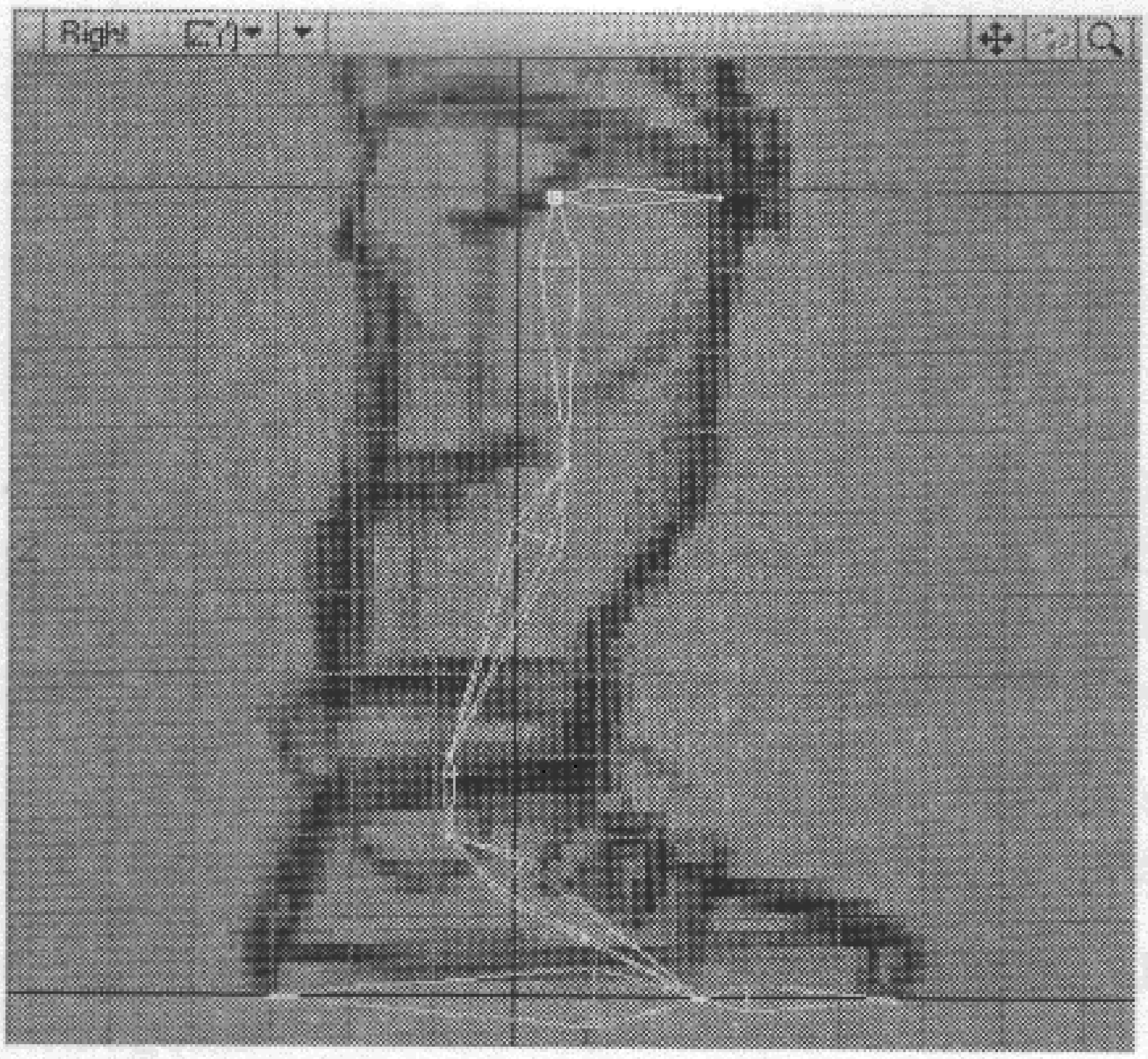
7 Use **Set Value** to place your entire leg chain at 100 mm on the X axis.

8 Switch to **Point** Selection Mode (CTRL+G) and select the two points in the rt Hip Bone.



9 **Set Value** to 495.3 mm on the Y axis.

10 Deselect the end point and **Set Value** to 22.7 mm on the Z axis.



11 Deselect all points and hit the \ key (**Display** > **Visibility** > **Unhide**). Now the base of the hip bone is perfectly aligned with the end of rt Hip Connector Bone.01.

12 Select **Tools** > **Points** **Merge** (M). Leave **Type** set to **Automatic** and click **OK**. One point should have been eliminated.

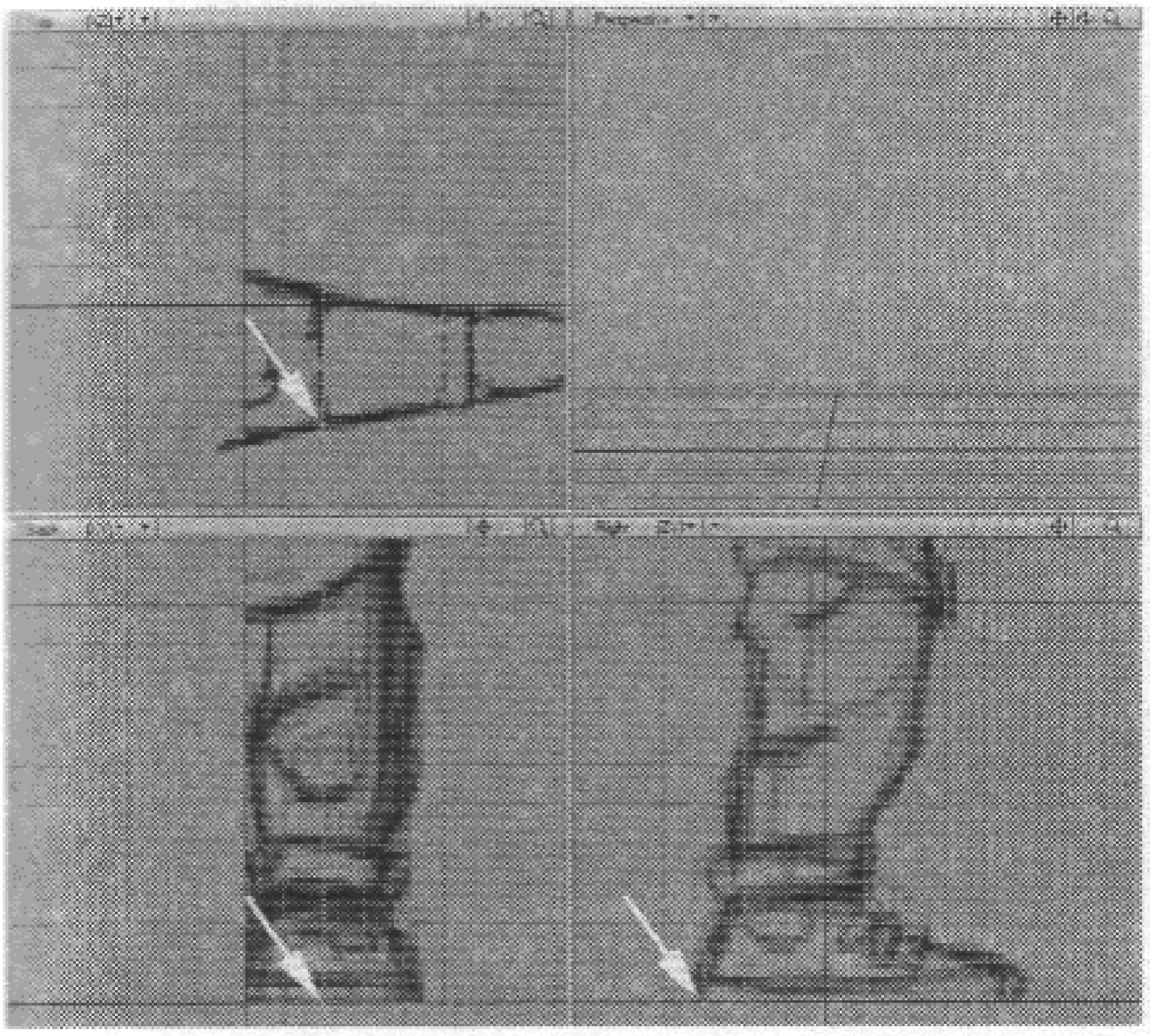
13 Select **Objects** > **File** > **Save Object As** and save this object as L6_Quincy_Legs.lwo.

Creating IK Goals in Modeler

A very important part of this IK setup is goal alignment. In order to maintain smooth rotations during animation and avoid IK *popping*, it is crucial that each initial goal position matches the corresponding bone pivot point location in world coordinate space. This process is much easier with Skelegons because we can copy the points that connect the Skelegons and paste them into new layers as nulls.

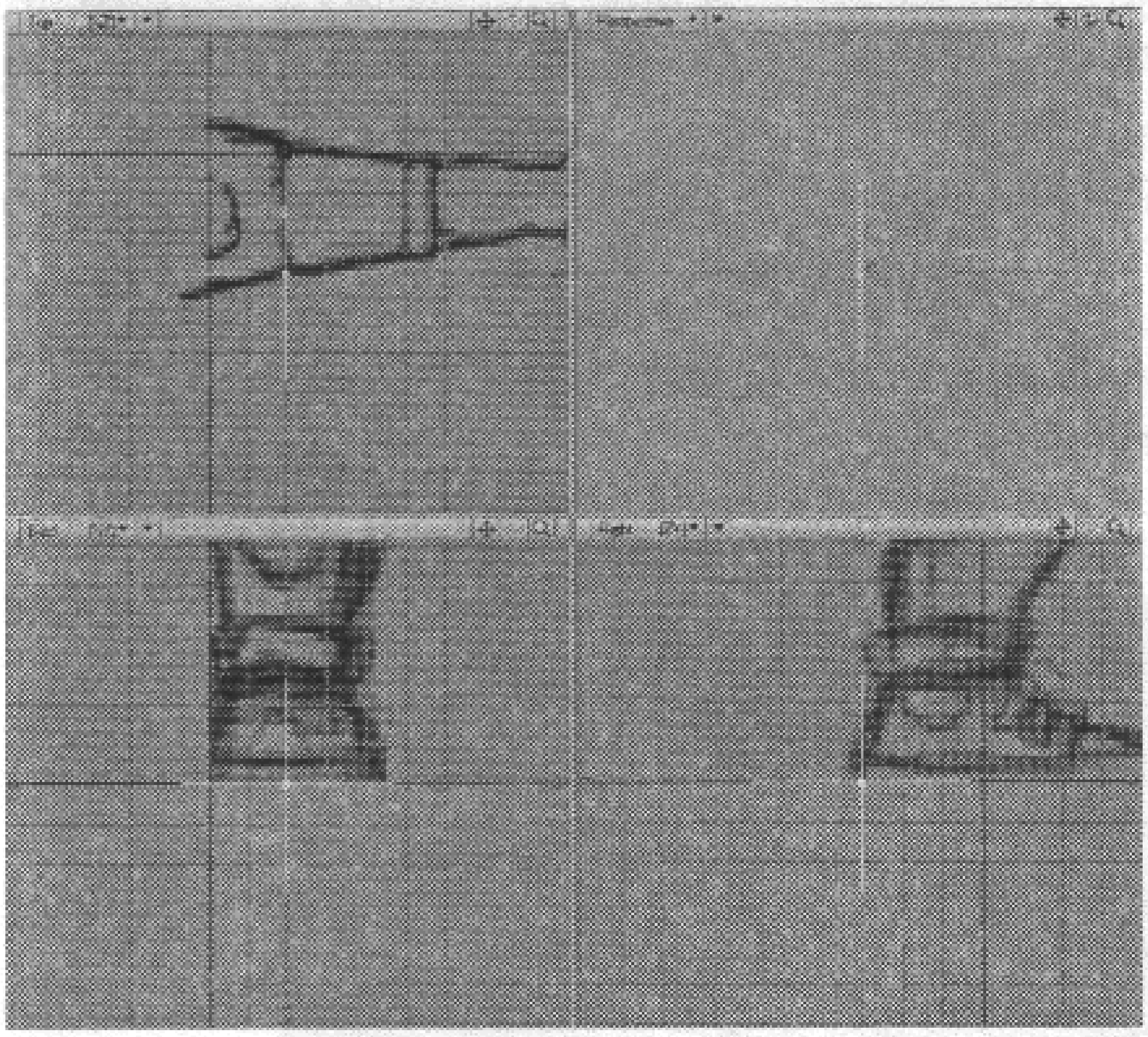
1 Select the point at the end of rt Heel Bone.00 and **Copy** it.

2 **Paste** the point into Layer 2 of L6_Quincy_Legs.lwo (your current object).



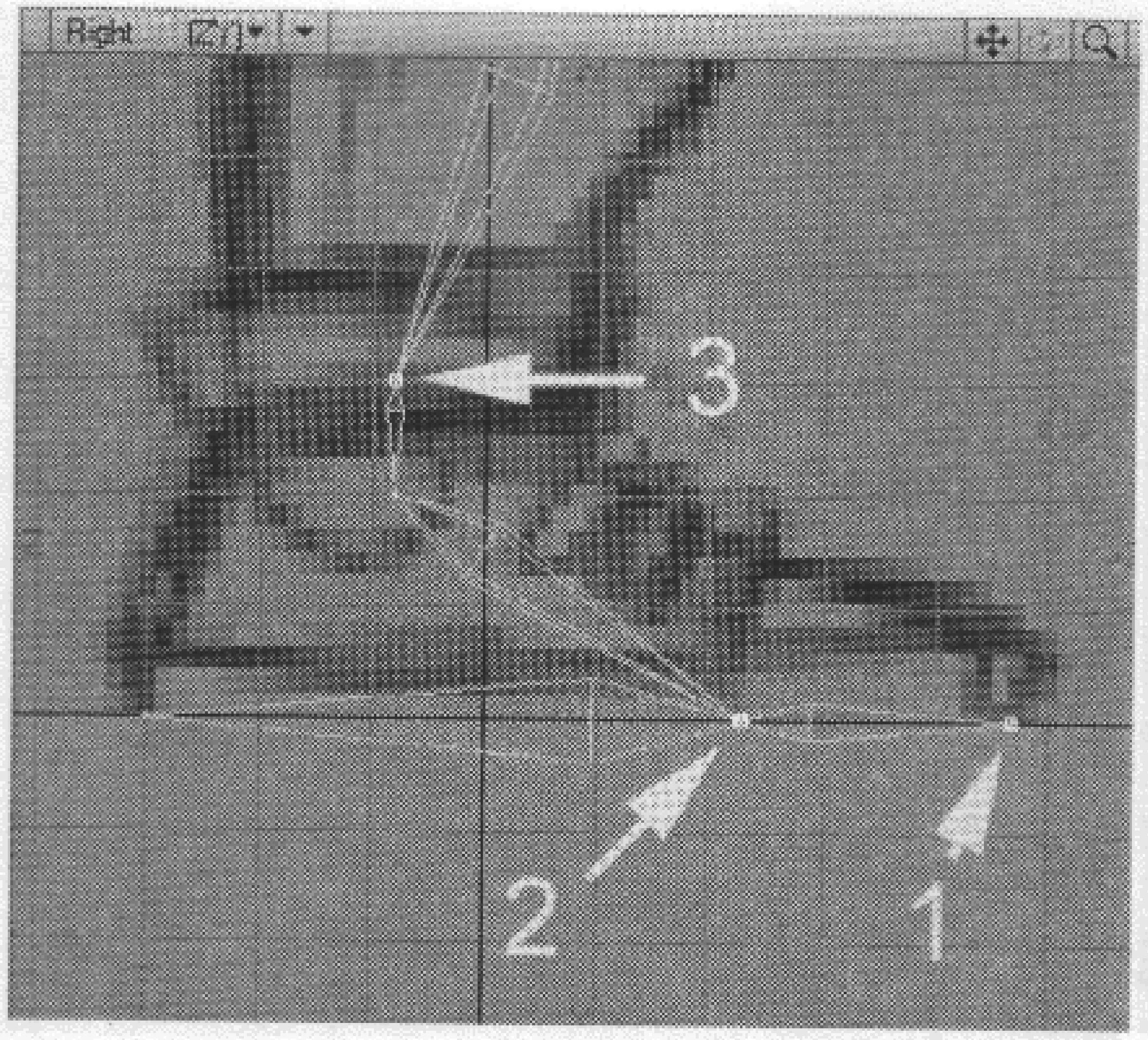
3 Press A (**Display** > **Fit Sel**) to center the point.

4 Select **Tools** > **Pivot**. A cyan crosshair appears at 0, 0, 0. This will act as the pivot point for this layer when it is loaded into Layout. Since the layer only has a single point in it, Layout will convert it to a null—ideal for use in our IK Goal hierarchy.



5 Click directly on the point. This should place the pivot directly over the point. Make sure the pivot is aligned with the point in all of the viewports. If you have difficulties aligning the pivot to the point zoom in to increase the Grid resolution, but do not turn off the **Grid Snap**. If you are still concerned (or just extremely obsessive), you can open **Tools** > **Selection Info** (I) and obtain the exact X, Y and Z coordinates for the point, then apply them to the pivot using its Numeric requester.

6 Repeat steps 1-5 of this section with the point at the tip of rt Toe Bone.01, the point joining rt Toe Bone.01 and rt Heel Bone.00 to rt Foot Bone, as well as the point at the start of rt Ankle Bone. Paste each point into its own layer in exactly the order given above. Be sure to also position the pivot for each layer.

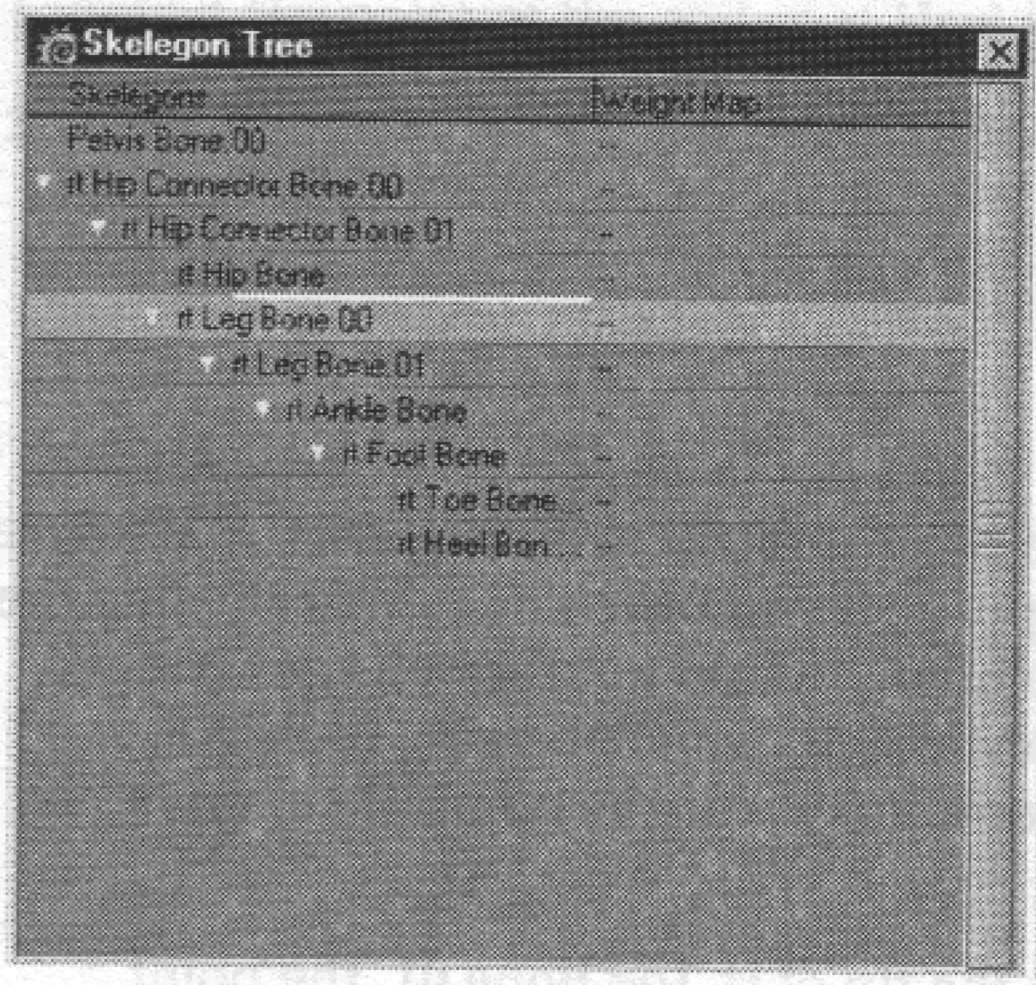


You should now have five layers. Layer 1 contains your Skelegons and layers 2-4 contain one point each. Each layer should have a different pivot location (leave Skelegon layer's pivot at 0, 0, 0).

Last Stop before Layout

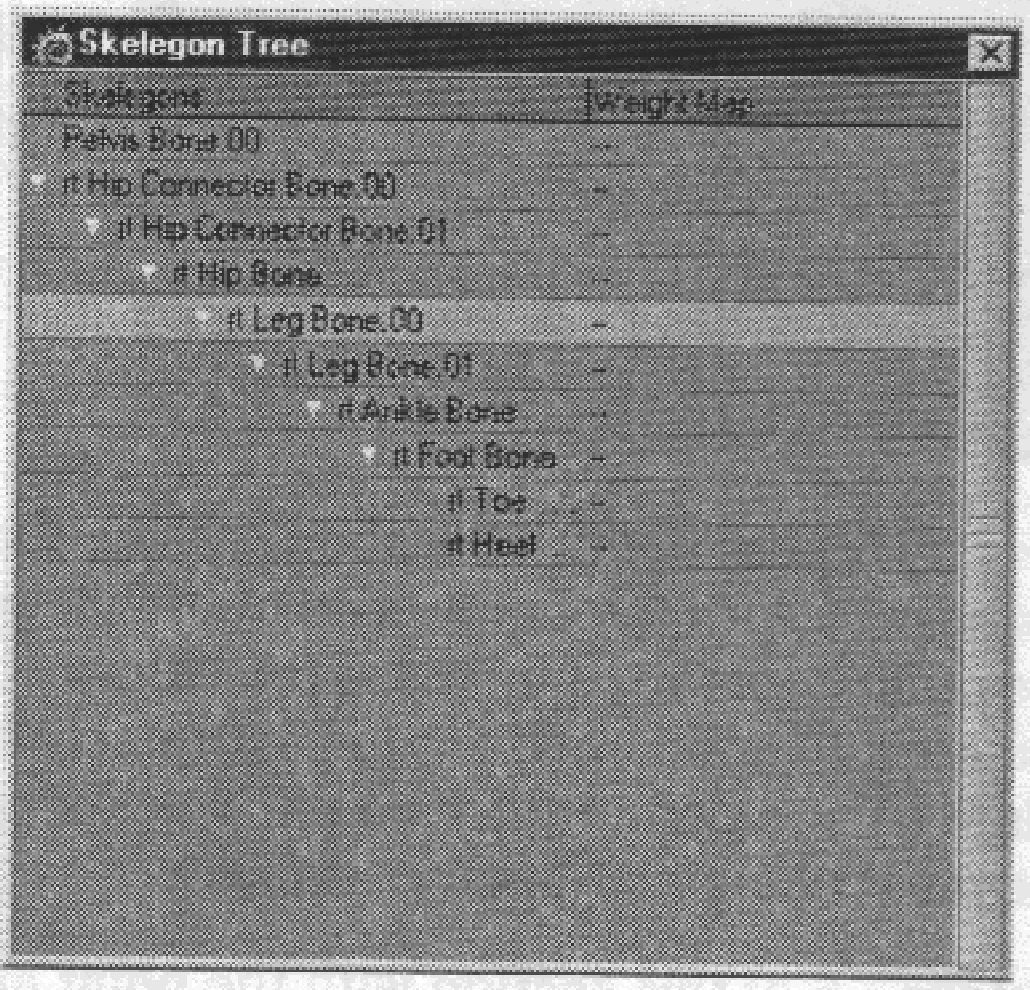
We're almost ready to enter Layout, convert our Skelegons to bones and setup IK. There are still a few minor details to attend to though. First up is final parenting. Although everything is exactly where we want it, the hierarchy is not right. If you open SkelegonTree you will see that rt Hip Connector.00 is not a child of Pelvis Bone and rt Leg Bone.00 is not a child of rt Hip Bone.00.

1 In the SkelegonTree panel, drag rt Leg Bone.00 up toward Pelvis Bone until the indented underline appears—signifying a parent-child relationship. Release the mouse button. Now rt Leg Bone.00 is a child of Pelvis Bone.



Skelegon Tree

Skelegons	Weight Map
Pelvis Bone.00	--
rt Hip Connector Bone.00	--
rt Hip Connector Bone.01	--
rt Hip Bone	--
rt Leg Bone.00	--
rt Leg Bone.01	--
rt Ankle Bone	--
rt Foot Bone	--
rt Toe Bone...	--
rt Heel Bon...	--



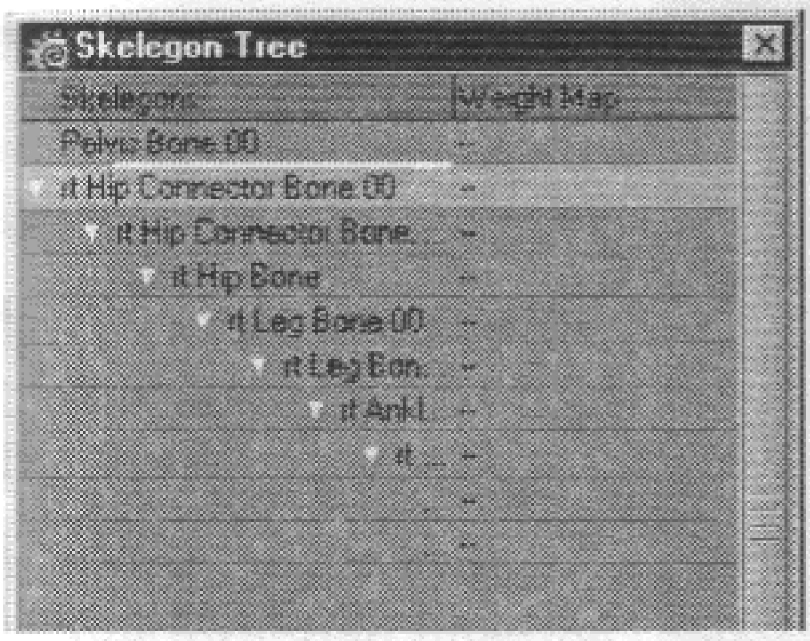
Skelegon Tree

Skelegons	Weight Map
Pelvis Bone.00	--
rt Hip Connector Bone.00	--
rt Hip Connector Bone.01	--
rt Hip Bone	--
rt Leg Bone.00	--
rt Leg Bone.01	--
rt Ankle Bone	--
rt Foot Bone	--
rt Toe	--
rt Heel	--

2 Close the SkelegonTree panel. The entire leg chain, including foot bones, moves forward on the Z axis as rt Leg Bone.00 snaps to the end of rt Hip Bone.00. Don't be alarmed;

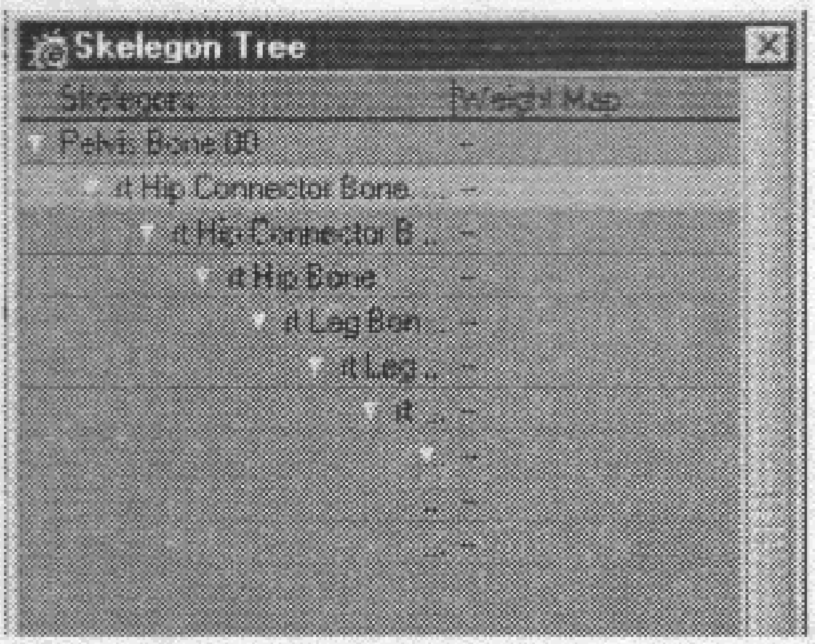
Skelegons must be connected end to end in order for one Skelegon to be the child of another. We will be able to change this by editing the affected bones in Layout.

3 Open the SkelegonTree once more and parent rt Leg Bone.00 to rt Hip Bone.00.



Skelegon Tree

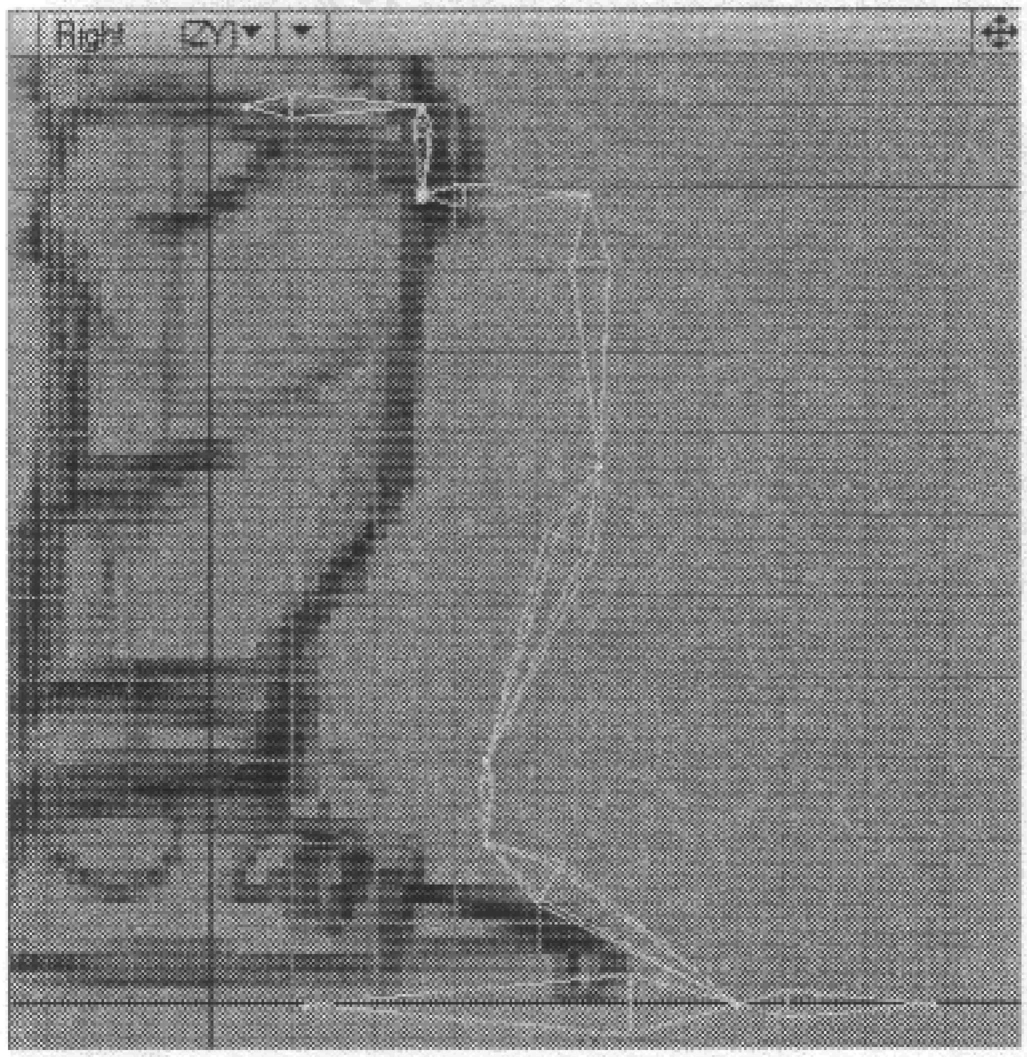
Skelegons	Weight Map
Pelvis Bone.00	--
▾ rt Hip Connector Bone.00	--
▾ rt Hip Connector Bone...	--
▾ rt Hip Bone	--
▾ rt Leg Bone.00	--
▾ rt Leg Bon...	--
▾ rt Ankl...	--
▾ rt ...	--
	--
	--



Skelegon Tree

Skelegons	Weight Map
▾ Pelvis Bone.00	--
▾ rt Hip Connector Bone...	--
▾ rt Hip Connector B...	--
▾ rt Hip Bone	--
▾ rt Leg Bon...	--
▾ rt Leg...	--
▾ rt ...	--
▾	--
...	--
...	--

4 Close the SkelegonTree panel. Now rt Hip Connector Bone.00 has snapped to the end of Pelvis Bone.00 moving the entire leg hierarchy again. Do not attempt to realign the goal points with the bones. Leave everything exactly where it is.

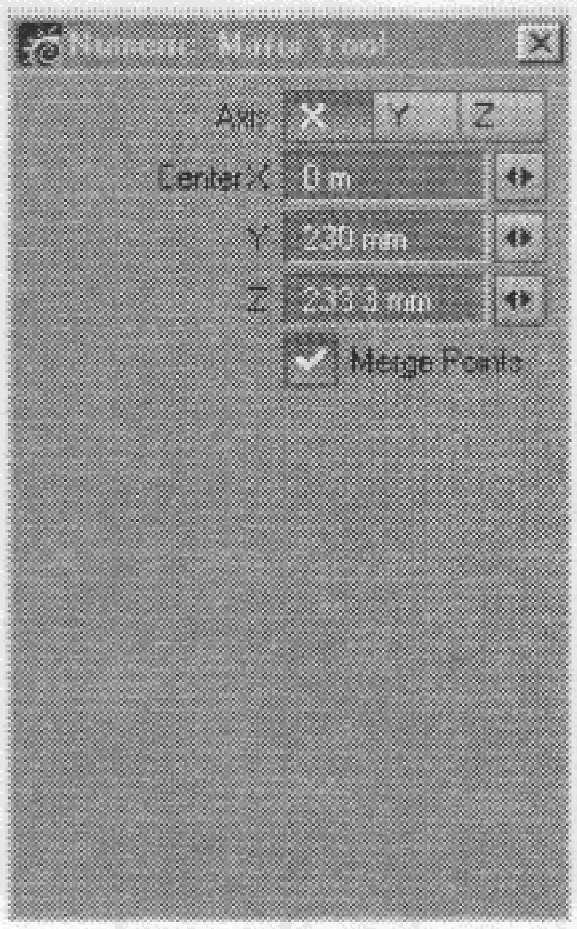


5 RMB drag a lasso around all of the Skelegons, except Pelvis Bone.00.

6 Click **Multiply** > **Replicate** > **Mirror** (SHIFT+V) to activate the **Mirror** tool.

7 In the Back viewport, position the cursor in the middle of the view over 0 on the X axis.

8 LMB click and drag straight up to define the mirror plane. Make sure to keep the cursor at X=0.



Numeric: Move Tool

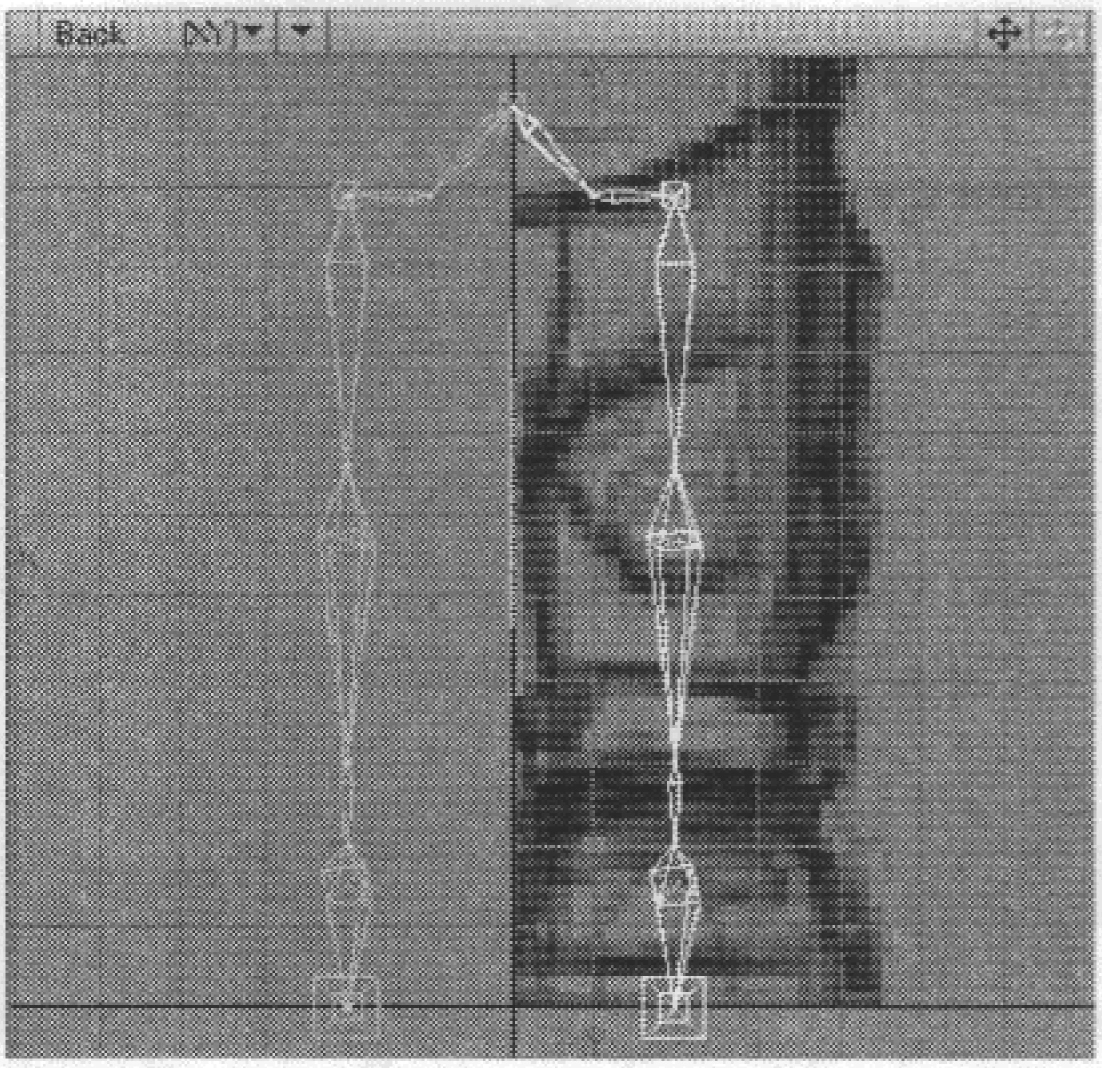
Axis: X Y Z

Center X: 0 m

Y: 230 mm

Z: 233.3 mm

☑ Merge Points

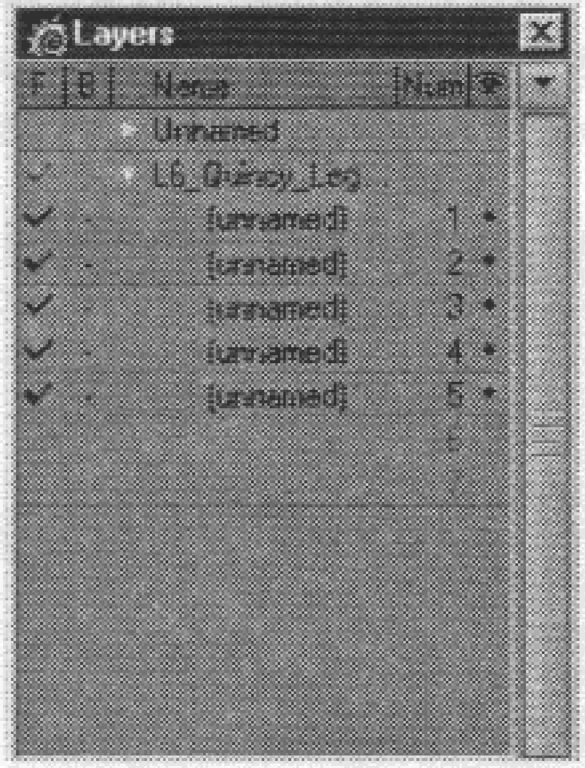


Now you have a matching left leg chain. If you open SkelegonTree, you will see two identically Skelegon hierarchies parented to the Pelvis Bone. Both have the rt prefix which, of course, is not right.

9 Double-click on each Skelegon in the second leg hierarchy and change its prefix to LF for Left.

10 Open the Layers panel (**Display** > **Layers**), if it's not already open.

11 Click on the white arrow to the left of L6_Quincy_Legs.lwo. The list expands displaying all five layers in the object. Currently, they are all unnamed.



Layers

F	B	Name	Num	◉
		Unnamed		
✓		L6_Quincy_Leg		
✓	-	(unnamed)	1	•
✓	-	(unnamed)	2	•
✓	-	(unnamed)	3	•
✓	-	(unnamed)	4	•
✓	-	(unnamed)	5	•

12 Double-click on the first layer in the panel. The Rename Layer panel pops up.

13 Type SKELEGON LEGS in the **Name** field and click **OK**.

14 Save your object.

Final Setup in Layout

1 Activate Layout.

2 Click **Actions** > **Add** > **Add Object** > **Load Object** and select L6_QUINCY_LEGS.LWO. You will see the nulls we created, but no Skelegons. As previously stated Skelegons must be converted to bones in Layout—they are merely placeholders for bones.

3 Select the L6_Quincy_Legs02.lwo:Skelegon Legs object. This is the layer we stored our Skelegons in. Notice that it loaded as an independent object.

4 Click **Settings** > **Bones** > **Cvt Skelegons**. A panel appears with the message *19 bones were created.*

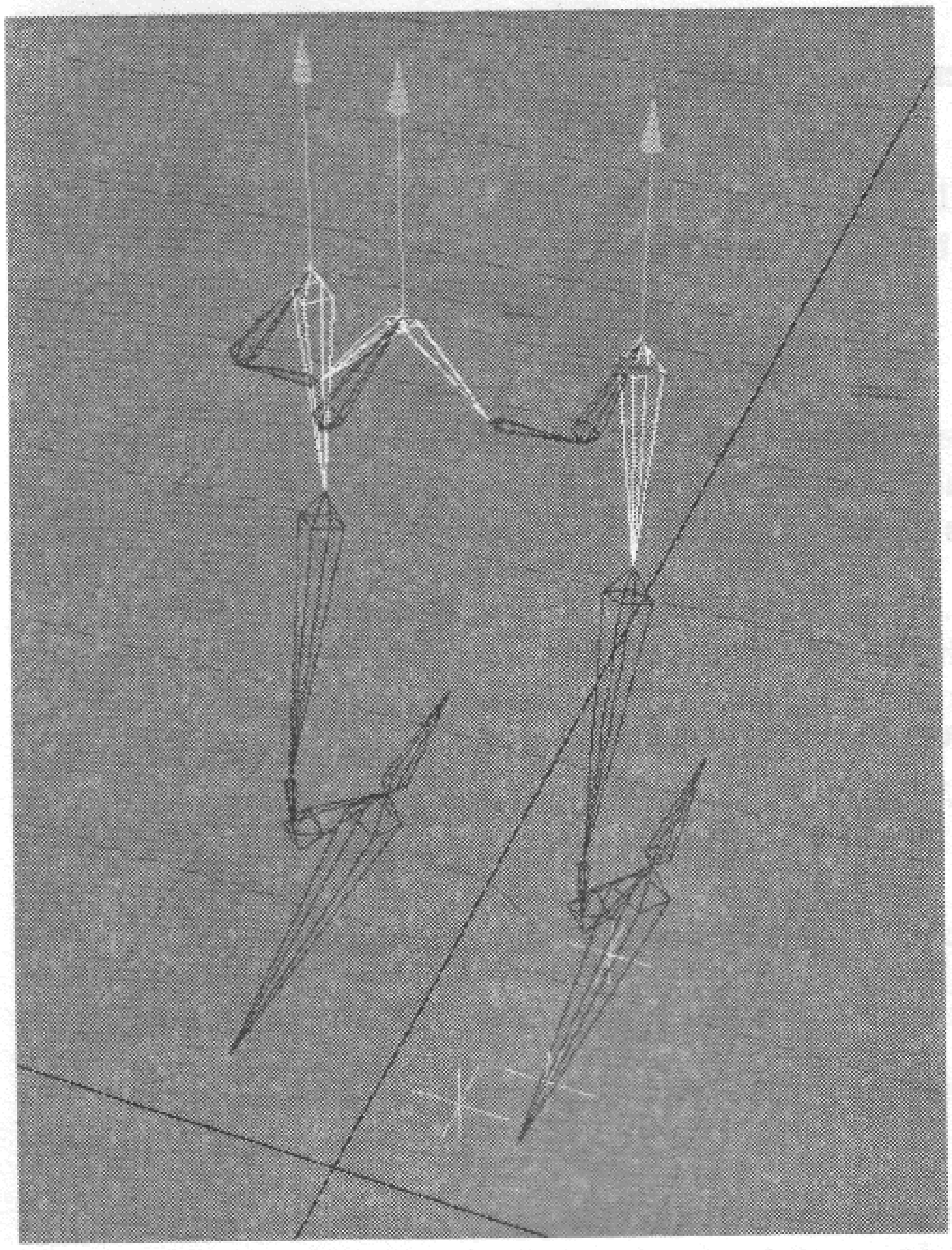
5 Click **OK** to close the panel. Now you'll see a set of bones which looks like our Skelegons. The first thing we will want to do is to remove the unwanted offsets generated by SkelegonTree.

1 Select rt Hip Connector Bone.00 and set its Move X, Y and Z value to 0. This restores it to our intended position.

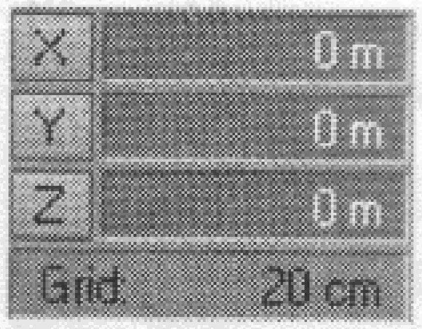
2 Create a keyframe for the bone's new position.

3 Hit the R key to reset the Rest Position for this bone to 0.

4 Repeat steps 1-3 for lf Hip Connector Bone.00, rt Leg Bone.00 and lf Leg Bone.00.



5 Turn off the Move X, Y, and Z channels for all bones accept for Pelvis Bone.00.



X 0 m
Y 0 m
Z 0 m
Grid 20 cm

Now let's setup the goal hierarchy. The first thing we should do is rename the Goals.

1 Open the Scene Editor.

2 Select L6_Quincy_Legs.lwo:Layer2.

3 Select **Actions** > **Replace** > **Rename Current Item**. You are presented with the Null Object Name panel.

4 Type rt Heel Null as the name and close the requester.

5 Repeat steps 1-4 for the remaining nulls. Change L6_Quincy_Legs.lwo:Layer3 to rt Toe Goal, L6_Quincy_Legs.lwo:Layer4 to rt Ball O Foot Null and finally L6_Quincy_Legs.lwo:Layer5 to rt Ankle Goal. Ignore the layer tags, they will be erased when the scene is saved because they are now nulls.

6 Select rt Ball O Foot Goal.

7 Select **Actions** > **Add** > **Clone Current Item**. The Clone Current Objects panel pops up. Click **OK** to accept the default of 1.

8 Parent rt Ankle Goal:Layer5 to rt Ball O Foot Null, rt Ball O Foot Null to rt Ball O Foot Null:Layer4, rt Ball O Foot Null:Layer4 to rt Toe Goal:Layer3 and rt Toe Goal:Layer3 to rt Heel Null:Layer2.

Now let's set up the IK channels

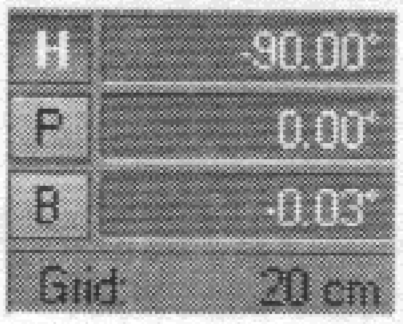
A wonderful new addition to Lightwave's impressive IK engine is the ability to set up IK on a per-channel basis. This allows you the freedom and flexibility of mixing Forward Kinematics with Inverse Kinematics, as well as any other channel specific modifier.

1 Select the rt Hip Connecter Bone.01. Choose **Actions** > **Motion Options** (M) and click **Unaffected by IK of Descendants** on the IK and Modifiers tab.

2 Use the down arrow on your keyboard to select the rt Hip Bone.

3 Click on the Controllers and Limits tab and select **Inverse Kinematics** from the **Bank Controller** pop-up menu.

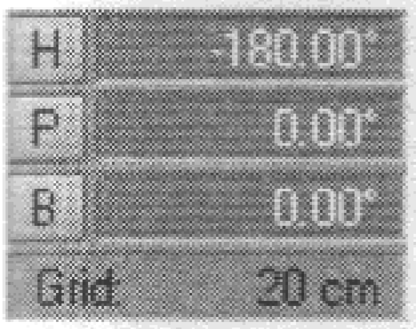
4 Disable the pitch and bank channels for rt Hip Bone. This will prevent us from accidentally rotating them during animation.



H -90.00°
P 0.00°
B -0.03°
Grid 20 cm

5 Use the down arrow to select rt Leg Bone.00 and set its **Pitch Controller** to **Inverse Kinematics**.

6 Disable all rotation channels. This will have no affect on the IK.



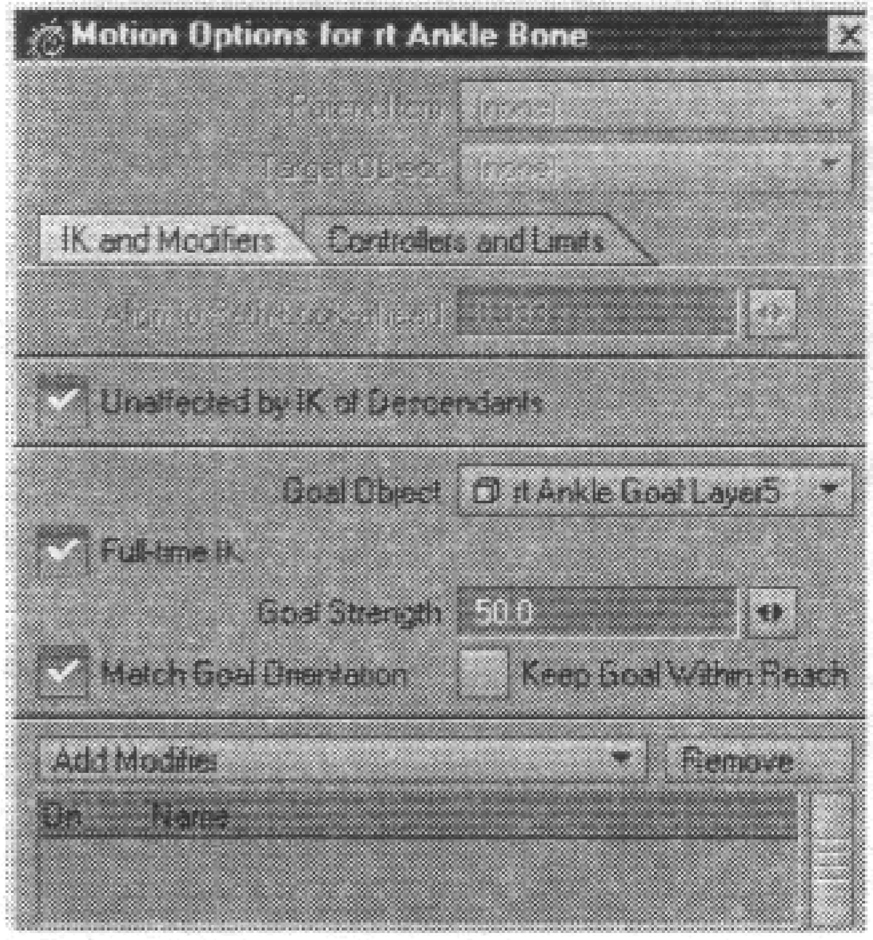
H -180.00°
P 0.00°
B 0.00°
Grid 20 cm

7 Do the same for rt Leg Bone.01.

8 Select rt Ankle Bone, click on the IK and Modifiers tab.

9 Select rt Ankle Goal:Layer5 from the **Goal Object** pop-up menu.

10 Select **Full-time IK**, **Match Goal Orientation**, **Unaffected by IK of Descendants** and set the **Goal Strength** to 50.0. These settings will end the leg's IK solution at the ankle and insure reliable foot placement during animation.



Motion Options for rt Ankle Bone

Parent Item (none)
Target Object (none)

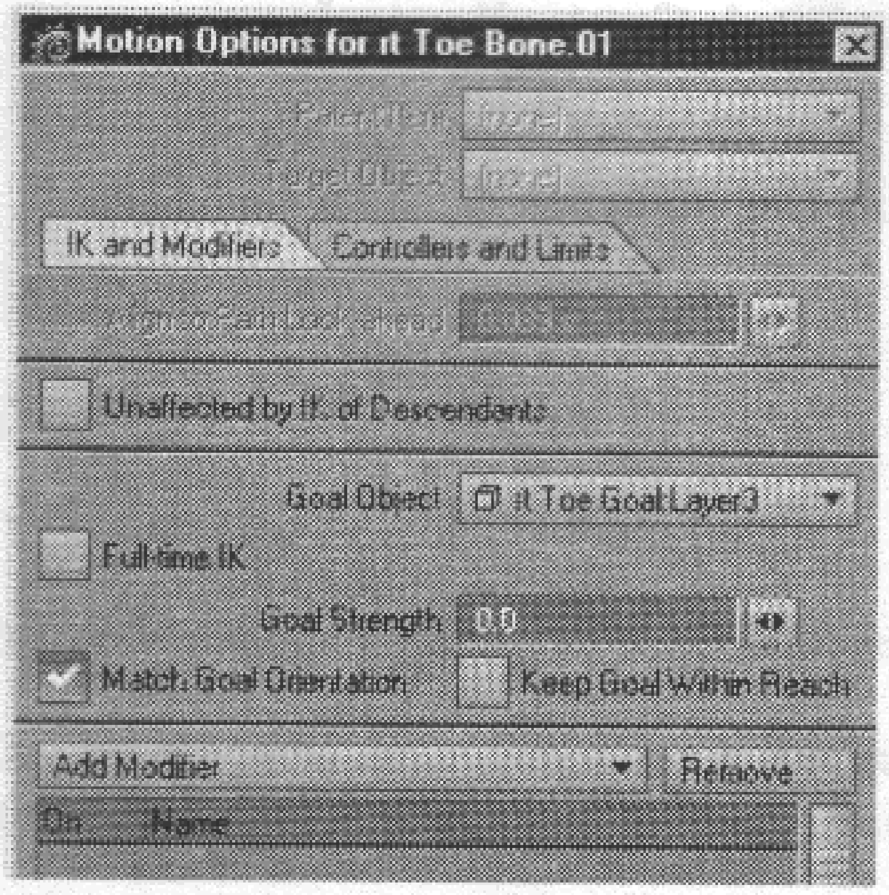
☑ Unaffected by IK of Descendants
Goal Object rt Ankle Goal Layer5
☑ Full-time IK
Goal Strength 50.0
☑ Match Goal Orientation □ Keep Goal Within Reach

Close the Motion Options panel so we can take look at what we've got so far. Right away you notice something's amiss: the right foot is now pointing straight up! This is because we turned on **Match Goal Orientation** for its Goal. This causes the bone to ignore any rotational data directly applied to it. Instead it matches the World Coordinate rotation of it's goal. To fix the foot alignment we need only rotate the Ankle Goal to match the original bone orientation.

11 Select rt Ankle Goal:Layer 5 and Rotate it 90.00° on its pitch axis. You don't have to worry about *gimbal lock* because we will not be directly manipulating this Goal.

12 Click on the rt Foot Bone pivot point then use the down arrow to select the rt Toe Bone.01.

13 Open the Motion Options panel again and select rt Toe Goal:Layer3 as the **Goal Object**. Leave **Full-time IK** turned off. Set the **Goal Strength** to 0.0 and turn on **Match Goal Orientation**.



Motion Options for rt Toe Bone.01

Parent Item (none)

Target Object (none)

IK and Modifiers | Controllers and Limits

Align to Path Look-ahead

☐ Unaffected by IK of Descendants

Goal Object: rt Toe Goal:Layer3

☐ Full-time IK

Goal Strength: 0.0

☑ Match Goal Orientation ☐ Keep Goal Within Reach

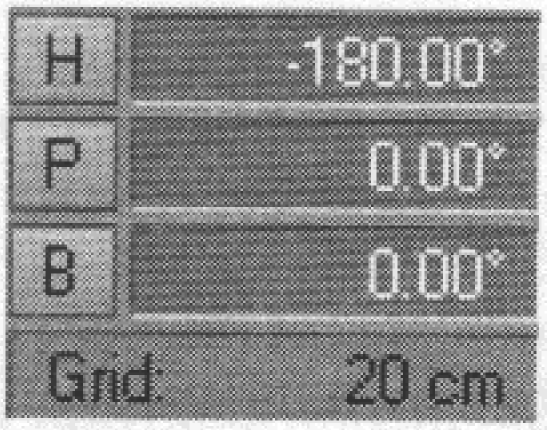
There's one final change we need to make before the right foot goal hierarchy is complete. We want to setup the rt Ball O Foot Null So that it is rotating in the opposite direction of the rt Toe Goal. Once this is setup rotating both rt Toe Goal:Layer3 and rt Ball O Foot Null:Layer4 simultaneously works nicely for *toe flopping* during walking or running animation.

14 Select rt Ball O Foot Null:Layer4 and **Rotate** it 180.00° on its Heading and create a keyframe. This will cause the ankle goal and the leg bones to be misaligned. This is where the Ball O Foot clone comes into play.

15 **Rotate** rt Ball O Foot Null (the clone of rt Ball O Foot Null:Layer4) -180.00° on its Heading and create a keyframe. This will compensate for the rotation of rt Ball O Foot Null:Layer4. We couldn't use **Record Pivot Rotation** here. It would have zeroed the rotation of rt Ball O Foot Null:Layer4 but left rt Toe Goal misaligned.

Tidying Up and Preparing to Animate

1 If it's not still selected, select rt Ball O Foot Null (the clone of rt Ball O Foot Null:Layer4) and disable all of its rotation channels.



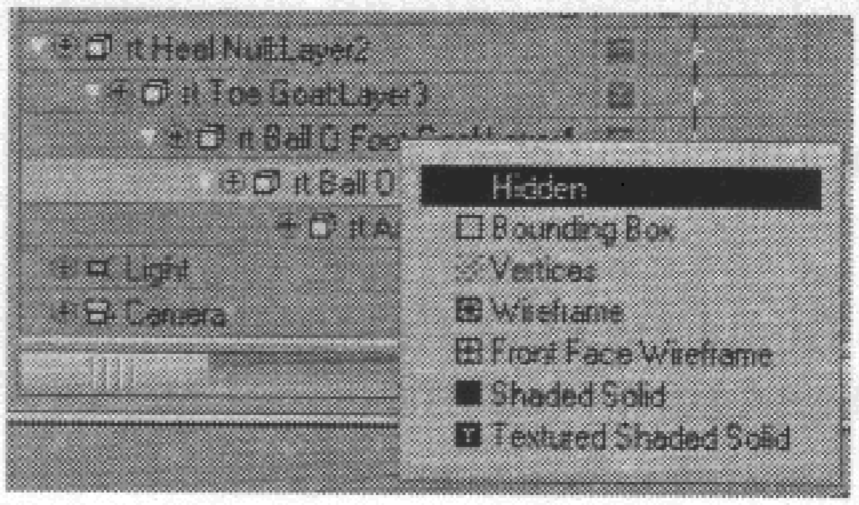
H -180.00°

P 0.00°

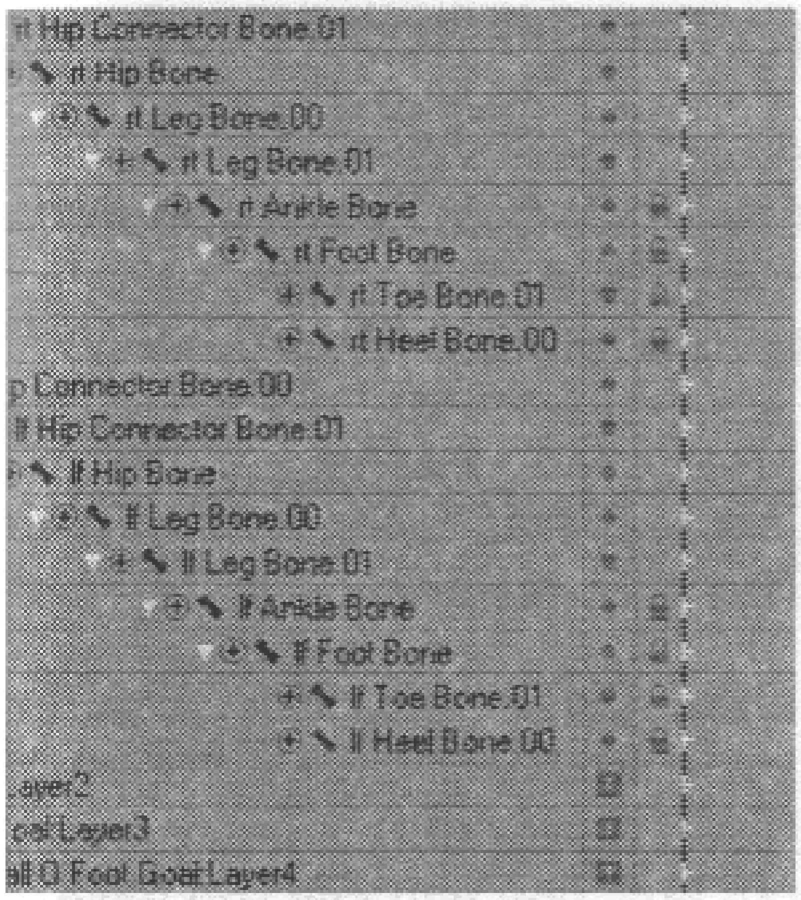
B 0.00°

Grid: 20 cm

2 In the Scene Editor set rt Ball O Foot Null to **Hidden**.



3 Lock all of the Bones in both feet. This will prohibit selecting them (via pivot point) with your mouse while still allowing you to see them for reference during animation.



Don't Forget the Left Foot

Of course, the left foot needs goals too. Instead of having you go through the entire foot goal hierarchy setup a second time, there is an already-built goal hierarchy for you to load. You will still have to make the IK goal assignment though.

1 Choose **Actions** > **File** > **Load Items From Scene** and load L6_QUINCY_LEFT_FOOT_GOALS.LWS.

2 Answer no to the query about loading lights from that scene.

3 Duplicate the IK settings from the right bone chain starting with lf Hip Connector Bone.01.

4 Don't forget to save your scene.

Once everything is in place, you can move and/or rotate the Pelvis Bone. Also try rotating the hip bones to *point* the knee.

USING GRADIENT SURFACES

There are many fantastic effects that can be created very quickly with Gradient surface parameters. This tutorial will give you a running start on using gradients to modify the look and feel of your surfaces!

1 Load the Gradient_Head.lws file from the Tutorials directory. Choose **Actions** > **Render** > **Render Options** and activate **Enable VIPER**. Hit the F9 key to render a frame. It is important to render a frame first so that VIPER can capture the extra buffer data to allow the interactive preview render.

2 Open the Surface Panel and choose the Body Surface and set the base color to blue. You will notice that VIPER updates with the new surface setting. Not very interesting is it?

3 Click the color Texture button to enter the texture editor and change the texture type to Gradient. You will notice that VIPER updates with a white surface as that is the condition of the gradient. It is important to understand that this gradient layer is still only a layer on top of the base surface color. If you change the layer opacity to 50% you will see the base blue merged back into the surface.

4 The first thing we want to do is change the input parameter to Incidence Angle. This will allow us to modify the surface color based on the surfaces angle to the camera. (See Gradient description in the Surfacing section of the manual for a more complete explanation of input parameters.)

5 Change the first key on the gradient to the blue that you used on the base color and then create another key by simply clicking on the bottom of the gradient strip. Set this key color to a nice light shade of blue. Mmmmmmm. Pretty.

6 Notice the nice soft shading effect you get when the color changes slightly across the angles of the surface. Use the Invert Keys command to give the surface more of a frosted appearance. Invert the keys once again to go back to the original shaded surface. Ooooh. It's so nice!

7 To intensify the effect a bit click on the gradient towards the top (dark blue area) to create a new key and then drag it down towards the lighter blue section at the bottom.

8 Now experiment with adding keys to the gradient and watch how much you can change that surface based on the incidence to the camera. Remember that as your object or camera moves the incidence will change and this will create animation of the surface. Very cool stuff!

9 Next change the Input Parameter to Light Incidence and pick the only light in the scene in the Light pop up beneath the Input Parameter.

10 If you drag the time slider in Layout you will see that this scene comes complete with some bad rotation values for the light! This will, however, provide you with a mechanism to view how the light's incidence angle to the surface can cause an animated surface. After you have updated the time slider click on Render on the VIPER palette. You should see the gradient surface change according to the new angle of light to surface. Get ready to really be impressed!

11 On the VIPER palette you will see a pop-up called Preview. Click on it and choose make preview. This will step through the frames and create an animated preview that you can

watch in real time after it is generated. With this in place you can see exactly how your surface will react to the light angle changes. Very cool.

☞ ***NOTE***

To use a light to change a gradient surface without effecting the surface illumination add a light and set intensity to zero. You can then use this light as a dedicated surface light modifying the surface based on a gradient reading the incidence of that "dark" light!

☞ ***NOTE***

You can reveal layers behind the gradients by using an Alpha value on some of the key frames. For an example load the Gradient_Alpha_Logo.lws from the Tutorials directory.

USING EXPRESSIONS

Expressions open up a whole new world of interaction in LightWave. Before we start, there are a few rules that you will want to remember when working with Expressions:

- Expressions are case sensitive. You need to pay special attention to what is capitalized and what is not. This is probably the most common mistake made when using expressions.

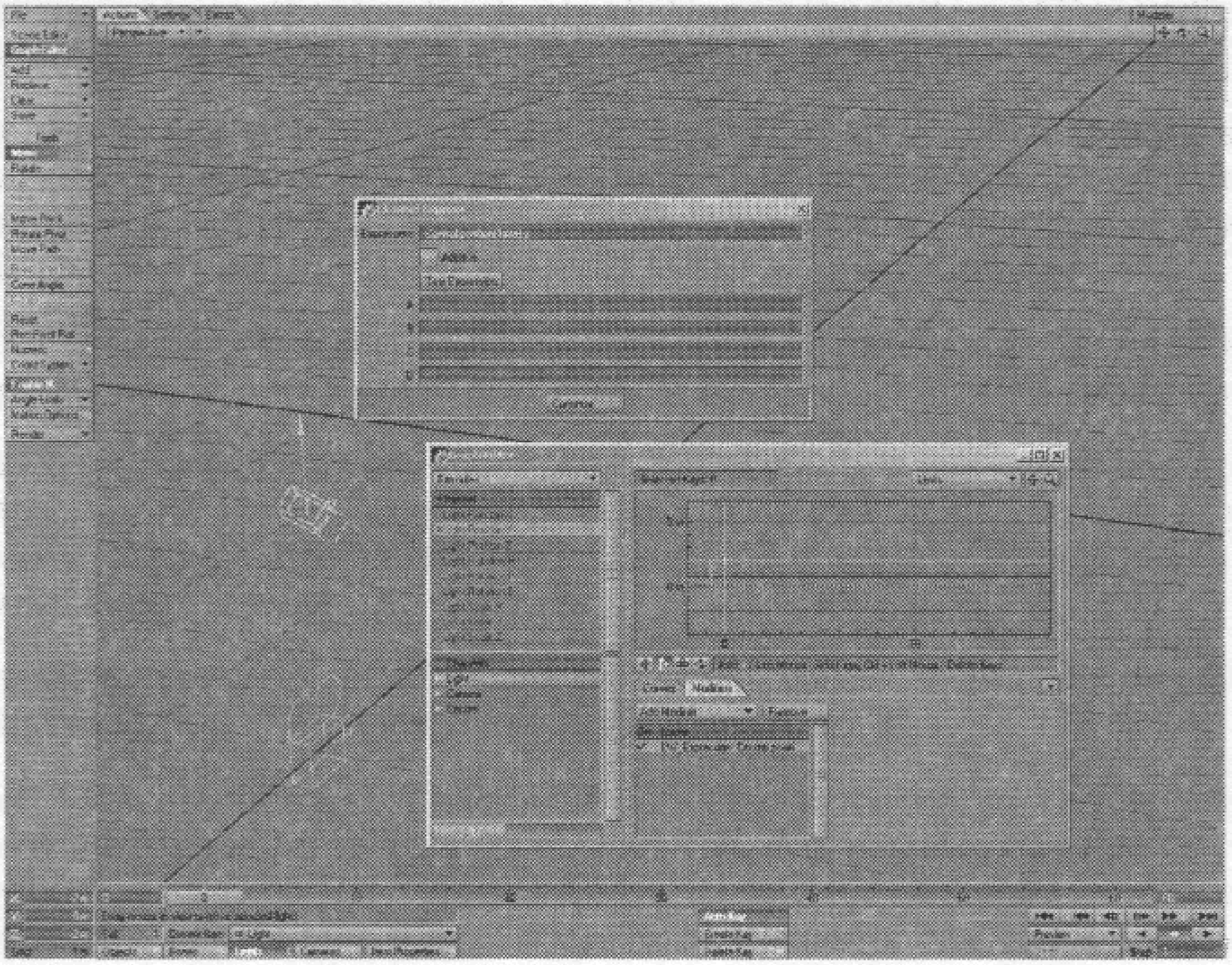
- Also, items cannot have spaces in their names. This means that you cannot reference something in LightWave that has been automatically named (i.e., Light (4), Null (2), etc.). To fix this, rename the item with a unique name (e.g., Bounce_Light4 or Hip_Goal).

Expressions Tutorial 1 - Following another item

Let's start off with the basics. In this example, we want the Light to always stay the same height (Y axis) as a null object. *Couldn't you just parent it?* Yes, you could, but that would force the light to move on all axes with the null object, we just want it to follow the null on its Y axis.

1 Add a null object and name it CONTROL. Remember, expressions are case sensitive, so be careful how you name your items. If you use a capital C here, your expression will also need to be capitalized.

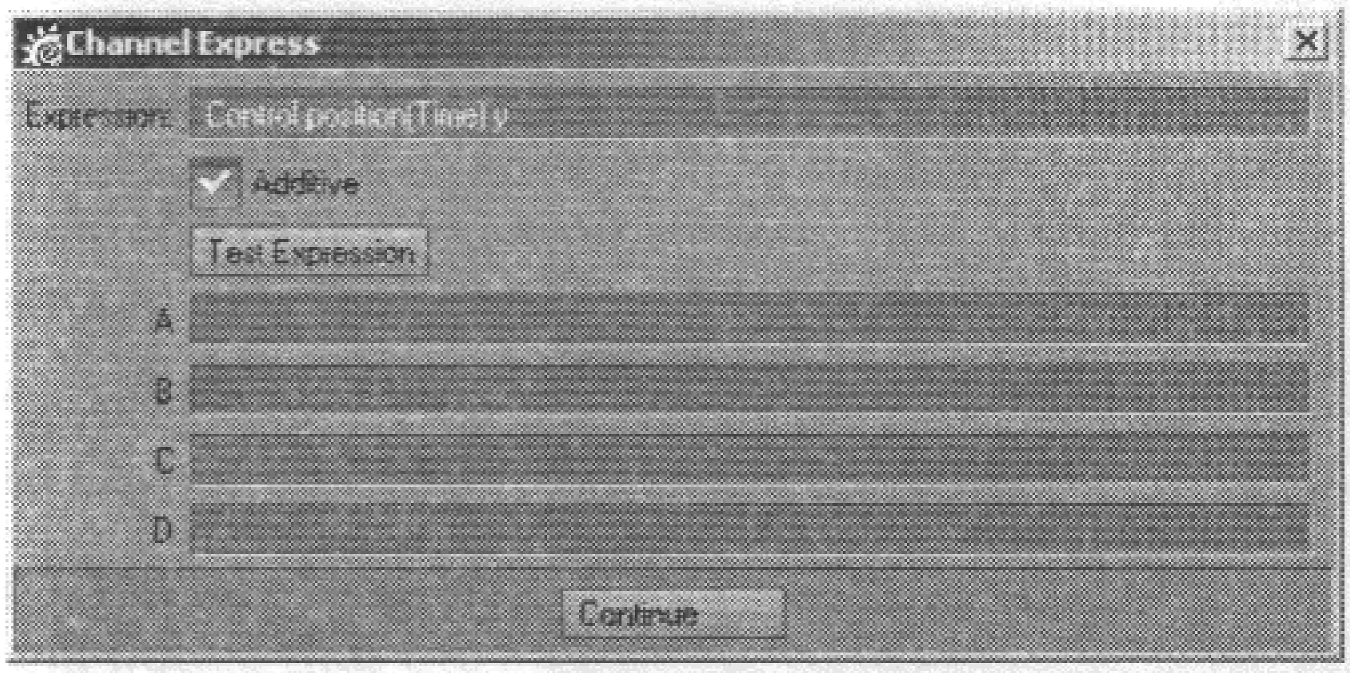
2 Click on the Light and open the Graph Editor. Expressions are channel based, and we want the Light to follow the null object on the Y axis, so click on the Light.Position.Y channel in the channel list on the left side of the Graph Editor.



3 Below the graph area of the Graph Editor are two tabs: Curves and Modifiers. An expression is considered a channel modifier. This simply means that any of the modifier plug-ins on this tab can modify the motion of the selected channel. Click on the **Add Modifiers** pop-up and select the Expression modifier.

4 Double-click on Expression in the Modifier list to open its panel. Enter the following into the Expression text area: Control.position(Time).y

Let's break this into it's parts to get a better understanding of what we've entered: *[Control].[position(Time)].[y]*. Now, *[Control]* is the item name in Layout that we are going to get the motion information from. *[position(Time)]* is the type of information that were trying to get from the item. This could also have been rotation or scale. *(Time)* refers to the current time (in seconds) in Layout. This information is returned as a *vector* (e.g., <0,1.5,0>) which requires a *channel selector* to strip out the correct channel data. *[y]* is the channel selector for the specific channel information.



Expression: Control.position(Time).y

☑ Additive

☞ **NOTE**

The shortcuts *pos* and *rot* are valid for position and rotation. scale is a shortcut for scale.....Ok, so it's not really a shortcut. If you wanted the Light to follow the null object on the X axis, you would add the following expression to the Light.Position.X channel: Control.position(Time).x

5 Click on **Test Expression**. If it gives you an error, check your capitalization for any differences from the equation above. If everything was fine, click **Continue**.

6 Close the Graph Editor and create a motion path for the null. Be sure to animate it on all axis. Play the animation. You should now notice that the Light follows the object whenever it moves on the Y axis, but doesn't follow it when you move on the X and Z axis. Look at it from different views for a better idea of what's going on in the scene.

7 Notice how the Light is locked directly to the object's position. The expression we entered ignores any of the keyframe motion of the Light. To add the motion from the expression to the motion of the Light, we need to turn on **Additive** mode. Select the Light and open the Graph Editor. Select the Light.Position.Y channel. Open the Expression panel and click the **Additive** checkbox to add the Lights keyframe motion to the expression. The Light should instantly jump up to it's original position. Close the panel.

☞ **NOTE**

Notice the dot to the left of the channel name in the Graph Editor's Scene List window? This tells you that a modifier is attached to this channel.

8 Play the animation again to see the difference that **Additive** mode makes to the Lights motion. If you want, animate the motion of the Light on the Y axis to see the additive effect.

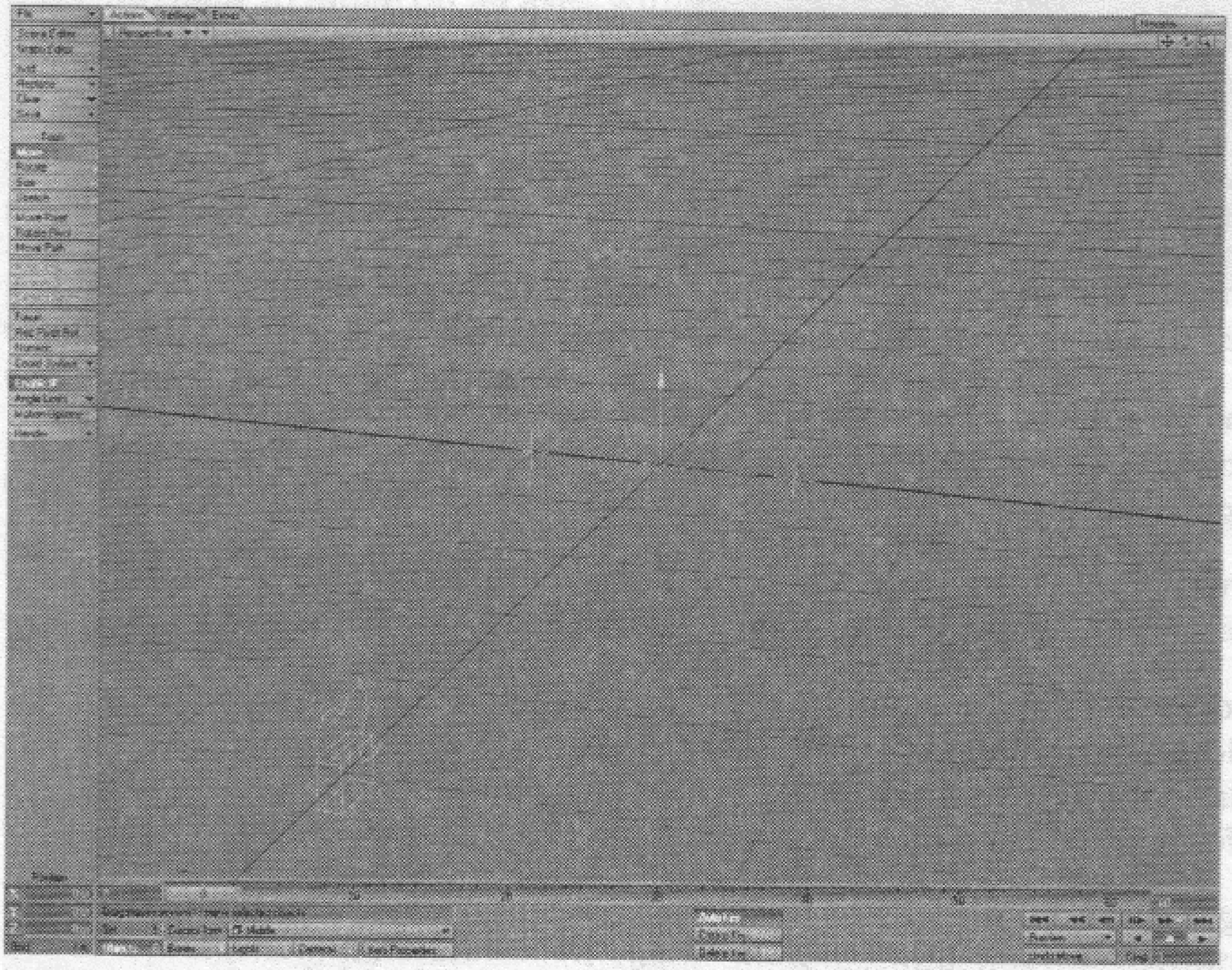
9 Congratulations! You've just written Follower! Seriously though, if you just wanted an item to behave like another item you could use follower, but the power of expressions allows you to do so much more.

Expressions Tutorial 2 - Stay in Between!

In this tutorial we are going to learn about using functions that are built into the expressions engine. We're going to write an expression that keeps an object centered between two other objects on the X and Z axis.

1 Clear the scene. Add three null objects and name them LEFT, RIGHT and MIDDLE. Remember to capitalize the names properly.

2 Move the Left null to the left about a meter and move the Right null to the Right about a meter. (Aren't these clever names!)



3 Select the Middle null and open the Graph Editor. Select the Middle.Position.X channel and add the Expression modifier to it.

4 Open the Expression panel and enter the following expression:

```
center( Left.position(Time) , Right.position(Time) ).x
```

Let's break this down into smaller parts. First there is the center function. It looks like this: *center(Item1 , Item2)*. The center function needs information from two different items. We are feeding it the position of the Left null and the position of the Right null. It needs the entire motion information for each object, so we don't add any channel selectors. After it calculates the center between those two items, we need to tell it which channel to give us as a result. We do this by adding the channel selector *.x* at the end of the center function.

☞ ***NOTE***

The center function will work for rotation and scale also.

5 Click on **Test Expression**. If it gives you an error, double-check to make sure it matches the expression above. In most cases it's a simple case of not capitalizing the object names, or the Time variable. Before you close the panel, highlight the entire expression and hit Ctrl+c on the keyboard. This copies the expression to the system's clipboard.

6 Close the Expression panel and select the Middle.Position.Z channel. Add the Expression modifier to that and open its panel.

7 Click and drag in the text entry area to highlight the *(none)* text. Hit Ctrl+v on the keyboard to paste the expression from the clipboard. Before we can use this expression, we need to make a change to it. In the previous expression, we were taking the X axis results from the center function, now we need the Z axis. Replace the .x with a .z The expression should now look like center(Left.position(Time) , Right.position(Time)).z

Test the expression and if everything works, close the panel and Graph Editor to get back to Layout.

8 Animate the Left and Right nulls on the X and Z axis. When you play the animation, you will notice that the Middle null now stays between them.

9 One thing you might have noticed is that the Middle null doesn't update it's position until you've created a keyframe. We can force Layout to update by telling the Middle null who it should depend on for an update. Select the Middle null and open the Motion Options panel.

10 Click on the **Add Modifier** pop-up and add the XDepend plug-in. Double-click to open the XDepend panel. Select the Left null in the **Depend on** pop-up. Add XDepend a second time and select the Right null. We've just told the Middle object to watch the Left and Right null objects for an update.

11 Close the panels to get back to Layout. Now when you move either of the other nulls, the Middle null should update its position. If it doesn't, you need to turn **Auto Key** on and from the options panel you should have **Auto Key Create** set to either **Modified Channels** or **All Motion Channels**.

12 Enjoy!

☞ ***NOTE***

This type of centering expression is often used as a starting point for the hips of a character. It allows the animator to focus on the animation of the legs and feet while the hips are controlled automatically.

Appendix

APPENDIX

OBJECT AND IMAGE FORMAT SUPPORT

LightWave generally supports the following object and image formats; however, depending which platform you have, all of the items listed below may not be available. Additionally, installed plug-ins can add make more formats.

Object Formats

In addition to loading and saving LightWave-format (.lwo) objects, the following formats are supported: 3D Studio, DXF and WaveFront. These file formats may be exported from Modeler using their respective *Export* commands in the **Additional** pop-up menu (e.g., Export_3DS). There is also a LightWave version 5.x object export command called **Export_LW5**, which will allow you to convert to the older object format.

Although you can also load these formats directly into Layout or Modeler, the objects will nearly always require some level of tweaking. Thus, it is recommended that you load them into Modeler first, make the necessary adjustments and then save as a native LightWave object before using in Layout.

☞ ***NOTE***

The 3D Studio format is the only (non-LightWave) conversion where some surface properties are preserved (color, double-sided, etc.).

Image and Animation Types

You can load and save various image and animation formats. What is available to you depends on the platform you are using, as well as the plug-ins you have active. Please check the NewTek web site (www.newtek.com) for additional plug-ins as they become available.

The **Flexible Format**, **TIFF LogLuv** and **Radiance** image formats contain high-dynamic-range data (discussed below). In addition, they store the data with floating-point accuracy instead of integer. This results in much more accurate image data.

☞ ***NOTE***

When a 32-bit format is selected for saved RGB images, the 8-bit Alpha image is stored along with the 24-bit RGB data. However, when saving alpha images directly, only the alpha image is saved, even if the format is 32-bit.

High Dynamic Range images (HDRI)

In computer graphics, color is displayed as a *triplet* value: red, green and blue. These values typically range from 0 to 255. Those 256 steps of color represent eight bits and together all three channels make up a 24-bit image. This means the maximum amount of color or luminance variation an image is allowed would be merely 256 steps.

☞ ***NOTE***

See the discussion on radiosity in the *Shadow and Light* chapter of the Shape manual for additional information on high-dynamic-range images.

In the real world, the human eye can perceive a much higher range of brightness and color values. Film can also react to a much wider range. Video cameras, however, are limited to a fixed range that fits closely to the same 256-step limit.

When exposed to *high-dynamic-range visuals*, such as a sunset or a desert landscape, the lens of a camera will produce some level of artifacting. Some of those artifacts can be seen as blooming areas of brightness, color bleed, luminance spill, lens streaking and many other *visual cues* that tell the viewer there is a very bright light source in the scene.

These very bright surfaces can also contribute to the overall lighting of a scene. For example, sunlight streaming into a room will bounce off the floor and add a subtle illumination to the walls and ceiling that would otherwise be left dark. All of these effects can be seen in images captured by devices that do not support high-dynamic-ranges.

Because computer graphics application were designed to output to devices that would not understand pixel values above RGB 255, 255, 255, most applications do not provide for any value to exceed these limits. With LightWave, this is changing forever. LightWave calculates all internal data without limits and with IEEE floating-point accuracy. This means that when LightWave points a light at a surface, while the final rendered pixel may only reach RGB 255, 255, 255 for pure white, internally that pixel may have reached ten times that amount. This may not seem significant at first glance—white is white after all—but, if we look at how LightWave utilizes that data, it becomes very exciting.

Import/Export

LightWave can utilize high-dynamic-range detail, as it is generated internally (e.g., in the case of a very bright light) or from data in image files. This can be imagery generated from a series of photographs taken at various exposures and composited (see *Recovering High Dynamic Range Radiance Maps from Photographs* by Paul E. Debevec and Jitendra Malik at HTTP://WWW.CS.BERKELEY.EDU/~DEBEVEC/RESEARCH/HDR/) or data rendered in LightWave saved in one of the high-dynamic-range formats.

Once these images are imported into the system, they can be used just like any other image in LightWave 3D(e.g., as a texture, background, etc.). During the rendering process LightWave will respect the extra data in the image to assist in secondary lighting and other calculations.

Imagine using a high-dynamic-range image as an environment wrap, which also illuminates the scene. With the appropriate imagery you can illuminate a scene without any lights and the results will match the look and feel of the original photograph.

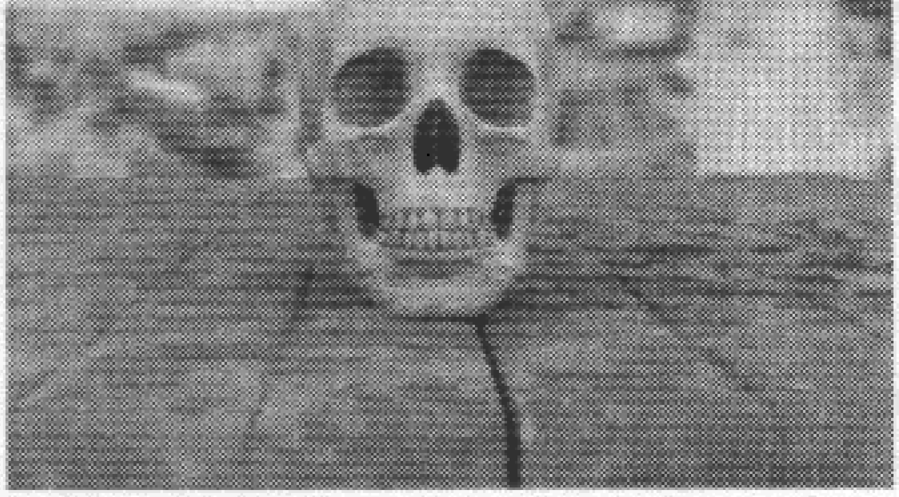


Image by Terrence Walker.

Once LightWave is finished rendering, images can be exported with the same high-dynamic-range data. This allows users to bring that data back into LightWave or into compositing applications that support such data. Using this extra data in the compositing process is very important as it can be used to more accurately represent imagery as it would look if it were being recorded directly to film. For example, compositing applications could use the extra dynamic range data to calculate the amount of diffuse bloom or color bleed from one pixel to the next.

Internal Compositing

Another area where high dynamic range imagery is supported is in LightWave's own internal compositing through pixel and image filters. Any filter can be designed to take advantage of the high-dynamic-range data with floating-point accuracy. This way, high-dynamic-range data can be leveraged in the post-process phase with included filters and by third-party additions.

CONFIGURATION FILES

The various LightWave configuration files are stored in your PROFILES directory under Windows NT. Generally, you will not need to edit these files directly. On the Macintosh, they are in the SYSTEM:PREFERENCES folder (with long names) and on Unix, they are in the user's home directory with a '.*rc' format.

LIGHTWAVE LIMITS

- unlimited points and polygons per object
- 1,023 points per polygon
- 4,096 bones per object
- 32,000 cameras per scene (see note)
- 32,000 objects per scene (see note)
- 32,000 lights per scene (see note)
- unlimited images
- unlimited surfaces

☞ **NOTE**

Because of memory considerations, the available number of cameras, objects and lights per scene is set at 100, 1000 and 1000, respectively, in the LW3.CFG file. The values may be increased—within the limits above, but not below 100; however, this will require more memory, so keep the new values as as small as possible. Do not edit the cfg file while LightWave is running.

MEASUREMENT UNITS

LightWave supports a large number of units of measurement. When entering numeric values into input fields, you may specify the unit of measurement using the following abbreviations:

Abbreviation	Description
um	Microns. One-millionth meter
mm	Millimeters. One-thousandth meter
cm	Centimeters. One-hundredth meter
m	Meters.
km	Kilometers. One thousand meters
Mm	Megameters. One million meters
mil	Mils. One-thousandth inch
in or "	Inches. 2.540 centimeters
ft or '	Feet. .3048 meters.
kft	Kilofeet. One thousand feet
mi	Miles. 5,280 feet
nmi	Nautical miles. 1.151 miles or 1852 meters

HINT

Since the **meter** is the default unit of measure for the **SI** or **Metric** unit systems, entering the meter equivalent is often easier than typing in the abbreviations. For example, enter .01 for centimeters, .001 for millimeters, 1000 for kilometers, etc.

Do the Math

Basic math functions are supported in numeric input fields. You can input a string such as "12ft+14m" and get an answer of "17.6576m."

MATH FUNCTIONS

Functions

abs(x)	absolute value
acos(x)	arccosine, return value in radians
acosh(x)	inverse hyperbolic cosine
asin(x)	arcsine, return value in radians
asinh(x)	inverse hyperbolic sine
atan(x)	arctangent, return value in radians
atan2(y,x)	arctangent of y/x, return value in radians
atanh(x)	inverse hyperbolic tangent
bessi(m,x)	Bessel function Im(x)
bessj(m,x)	Bessel function Jm(x)
bessk(m,x)	Bessel function Km(x)
ceil(x)	round up
cos(x)	cosine, x in radians
cosh(x)	hyperbolic cosine
dbessi(m,x)	derivative of Bessel function: Im'(x)

dbessj(m,x)	derivative of Bessel function: Jm'(x)
dbessk(m,x)	derivative of Bessel function: Km'(x)
djroot(m,n)	nth non-zero root of Jm'(x)
exp(x)	e (2.718..) raised to the power of x
fact(n)	factorial (n!)
floor(x)	round down
jroot(m,n)	nth non-zero root of Jm(x)
ln(x)	natural logarithm (base e)
log(x)	logarithm to the base 10
noise(x,y,z)	Perlin noise
rand(n)	Random value, seed n
sgn(x)	-1 if x<0, 0 if x=0, +1 if x>0
sin(x)	sine, x in radians
sinh(x)	hyperbolic sine
sqrt(x)	squareroot
tan(x)	tangent, x in radians
tanh(x)	hyperbolic tangent

Constants

_acres_per_sq_km	247.1
_air_density	1.293
_air_mol_mass	.02897
_atm_per_psi	.06804
_avagadro	6.0220e23
_boltzmann	1.3807e-23
_c	2.997925e8
_cm_per_in	2.54
_coulomb_const	8.98755e9
_deg_per_rad	57.2958
_earth_esc_spd	1.12e4
_earth_grav	9.80665
_earth_mass	5.98e24
_earth_radius	6.37e6
_earth_to_moon	3.844e8
_earth_to_sun	1.496e11
_eps0	8.85419e-12
_erg_per_joule	1e7
_eulers_const	.577215664901532860 61
_ft_per_m	3.280839895
_g	6.672e-11
_gas_const	8.314
_gauss_per_tesla	1e4
_gm_per_oz	28.34952313
_golden_ratio	1.6180339887498948482
_h	6.6262e-34
_hbar	1.05459e-34
_joule_per_btu	1054.35
_joule_per_cal	4.184
_joule_per_ftlb	1.356
_joule_per_kwh	3.6e6

_kg_per_slug	14.59
_km_per_mi	1.609344
_knots_per_mph	.86897624
_lbs_per_kg	2.204622622
_lit_per_gal	3.785411784
_me	9.1095e-31
_mn	1.67495e-27
_moon_grav	1.62
_moon_mass	7.35e22
_moon_period	2360448.
_moon_radius	1.738e6
_mp	1.67265e-27
_mu0	1.256637e-6
_oz_per_gal	128.
_pasc_per_atm	101325.
_pasc_per_psi	6895.
_pasc_per_torr	133.32
_pi	3.14159265358979323846
_qe	1.60219e-19
_solar_const	1350.
_speed_sound	331.
_sun_mass	1.99e30
_sun_radius	6.96e8
_watts_per_hp	745.712
_zero_deg_cels	273.15

Other

ifeq(a,b,t,f)	if-equal, returns t if a equals b otherwise returns f
iflt(a,b,t,f)	if-less-than, returns t if a is less than b otherwise returns f
ifgt(a,b,t,f)	if-greater-than, returns t if a is greater than b otherwise returns f
iflte(a,b,t,f)	if-less-or-equal, returns t if a is less than or equal to b otherwise returns f
ifgte(a,b,t,f)	if-greater-or-equal, returns t if a is greater than or equal to b otherwise returns f

COLOR CHART

Blacks	R	G	B
black	0	0	0
ivory black	41	36	33

Greys	R	G	B
cold grey	128	138	135
grey	192	192	192
slate grey	112	128	144
warm grey	128	128	105

Whites	R	G	B
antique white	250	235	215
azure	240	255	255
bisque	255	228	196
blanch almond	255	235	205
cornsilk	255	248	220
eggshell	252	230	201
floral white	255	250	240
gainsboro	220	220	220
ghost white	248	248	255
honeydew	240	255	240
ivory	255	255	240
linen	250	240	230
navajo white	255	222	173
old lace	253	245	230
seashell	255	245	238
snow	255	250	250
wheat	245	222	179
white	255	255	255
white smoke	245	245	245

Reds	R	G	B
brick	156	102	31
cadmium red	227	23	13
coral	255	127	80
firebrick	178	34	34
indian red	176	23	31
maroon	176	48	96
pink	255	192	203
raspberry	135	38	87
red	255	0	0
salmon	250	128	114
tomato	255	99	71

Oranges	R	G	B
cadmium orange	255	97	3
carrot	237	145	33
orange	255	128	0
orange red	255	69	0

Yellows	R	G	B
banana	227	207	87
cadmium yellow	255	153	18
dougello	235	142	85
forum gold	255	227	132
gold	255	215	0
goldenrod	218	165	32
melon	227	168	105
yellow	255	255	0

Browns	R	G	B
beige	163	148	128
brown	128	42	42
burnt sienna	138	54	15
burnt umber	138	51	36
chocolate	210	105	30
flesh	255	125	64
khaki	240	230	140
rosy brown	188	143	143
raw sienna	199	97	20
raw umber	115	74	18
sepia	94	38	18
sienna	160	82	45
saddle brown	139	69	19
sandy brown	244	164	96
tan	210	180	140

Blues	R	G	B
blue	0	0	255
cobalt	61	89	171
dodger blue	30	144	255
indigo	8	46	84
jackie blue	11	23	70
manganese blue	3	168	158
midnight blue	25	25	112
navy	0	0	128
peacock	51	161	201
powder blue	176	224	230
royal blue	65	105	225
slate blue	106	90	205
sky blue	135	206	235
steel blue	70	130	180
turquoise blue	0	199	140
ultramarine	18	10	143

Cyans	R	G	B
aquamarine	127	255	212
cyan	0	255	255
turquoise	64	224	208

Greens	R	G	B
chartreuse	127	255	0
cobalt green	61	145	64
emerald green	0	201	87
forest green	34	139	34
green	0	255	0
lawn green	124	252	0
lime green	50	205	50
mint	189	252	201
olive drab	107	142	35
sap green	48	128	20
sea green	46	139	87
spring green	0	255	127
terre verte	56	94	15

Magentas	R	G	B
blue violet	138	43	226
jasoa	160	102	211
laker purple	153	51	250
magenta	255	0	255
orchid	218	112	214
plum	221	160	221
purple	160	32	240
violet	143	94	153