GNOMON SCHOOL PRESENTS

HIGHEND MAGAZINE

ISSUE NUMBER 2 • SPRING 2003

Kung Pow
ENTER THE FIST
a compositing confessional

Demonslayer

Lexus' Minority Report

Digital Doubles
FOR VISUAL EFFECTS

Creating Effects with Camera Projection

ISSN 1541-5864
02
0 74470 97212 9
ISSUE 2 $12.95 USD

GNOMON

School of Visual Effects

Specialized training for careers in high-end computer graphics for the entertainment industries.

323.466.6663

www.gnomon3d.com

- Courses on the tools and techniques of the digital visual effects industry
- Courses on fundamental skills such as production design and miniatures
- 21-month certificate program, extension courses or custom training
- 9-Week intensive Maya Fast Track Program
- Over 15 individual hands-on Maya classes
- Classes held on evenings and weekends
- Maya DVDs, videos and tutorials from the Gnomon Workshop
- Maya streaming-video training and forums at Gnomon Online

"I can't say enough good things about my experience at Gnomon. Not only did the Certificate Program help me cultivate the necessary skills to make that happen, it provided tremendous focus and perspective and gave me the confidence to set my goals high. The program is truly first-rate, as are the members of the Gnomon faculty who are well-connected, successful industry professionals who have a passion for inspired education."

Steeg Haskel, Associate Cinematic Artist
Blizzard Entertainment

Gnomon Inc. School of Visual Effects
1015 North Cahuenga Blvd.
Hollywood, CA 90038
323.466.6663
www.gnomon3d.com

REALVIZ Adobe VISUALIZE hemisphere Maya

CONTRIBUTORS

DARIUSH DERAKHSHANI is a CGI effects animator on staff with Sight Effects in Venice, Ca. where he animates special effects for award winning national television commercials. He has shared top honors from the AICP and London International Advertising Awards.

JOHN LAFAUCE JR.– A respected compositor, John has been a "go to" guy for tackling difficult effects shots and producing quality results for many years. He is an active member of the Visual Effects Society and was recently selected to be a judge on the "Best Compositing" nomination panels for the first annual Visual Effects Society Awards.

ED HARRISS is a CG artist living in the USA. Currently, he runs a CG-oriented website dealing mostly with mental ray, SOFTIMAGE | XSI, SOFTIMAGE | 3D, some of his personal work (http://www.EdHarriss.com). He works for a company called Alternate Route Studios. (http://www.altroutestudios.com)

CRAIG SYLVESTER grew up in the three most populous cities in the world: Los Angeles, Beijing and Cairo. After a lifetime of studying art, indulging a fascination with computers full-time since the age of 15 and with CGFX since 18, he decided that the only thing left for him was to breathe special effects for film like a fish does water. He received a CGFX degree from the Savannah College of Art & Design in 2000, enjoyed a fantastic effects house internship, learned a lot about teamwork and the FX process from script-to-screen on the "Demonslayer" project, and continues to pursue effects projects from his desk in North Hollywood.

KIRK SHINTANI is an animator currently finishing the Certificate program at Gnomon School of Visual Effects. Prior to Gnomon, he graduated from UCLA with a psychology degree, and dabbled in Theater and Film before focusing his attention on 3D.

PAM HOGARTH, Director of Industry Relations at Gnomon, Inc., has been in the world of computer graphics for over 19 years. In that time she has done marketing, public and industry relations, and training for a wide range of companies and educational institutions. She has taught at The American Film Institute, Otis College of Art and Design, and Digital Media Institute (where she was Director) and has lectured on computer graphics, digital careers and visual effects.

KAT CURRY is a student and lab tech at Gnomon School of Visual Effects, with a specialization in Character Set Up and Particles and Dynamics. Before entering the program at Gnomon, Kat was a CG Programmer at Boston University, working in scientific visualization. She had previously held numerous positions as a programmer and interface designer at small companies.

SEAN MILLS, experiences as Lead Modeler at Zoic Studios Projects include creature and digital double modeling & texturing work for "Buffy The Vampire Slayer," "Angel," "Smallville," and the upcoming "NYPD 2069." He has also worked on music videos for Linkin Park, Garbage, and Mudvayne and commercial spots for United States Marine Corps, Dr Scholls Footwear, and Stolis Citrona.

MATT PLEC was introduced to digital visual effects at USC's Computer Animation Lab and, hooked for good, has been working in post production and related software development ever since. He is currently responsible for 2D/compositing technologies at ESC Entertainment in Alameda, California and can be reached via email at mplec@highend2d.com.

DARRIN KRUMWEIDE Associate Director, Gnomon School of Visual Effects Technical Editor for "Highend Magazine" Contributor of training media for The Gnomon Workshop and Gnomon Online and various publications. Graduate of Art Center College of Design with a B.S.in Industrial Design. Former Application Engineer for Alias | wavefront, consulting for studios and game companies throughout the Western US. Freelance and employed digital artist for film and video game production studios in Los Angeles.

Writers — Kat Curry • Dariush Derakhshani • Ed Harriss • Pam Hogarth • John Lafauce, Jr. • Darrin Krumweide • Sean Mills • Matt Plec • Kirk Shintani • Craig Sylvester

CALL FOR WRITERS
If you think you are qualified to write for Highend Magazine, e-mail us at: **writers@highendmag.com**

GNOMON SCHOOL PRESENTS
HIGHEND MAGAZINE

ISSUE NUMBER 2 • SPRING 2003

Publishers
CHARLES A. EDGIN charles@dmgpublishing.com
ALICE M. EDGIN alice@dmgpublishing.com

Editors
WILL MCCULLOUGH will@highendmag.com
ALEX ALVAREZ alex@highendmag.com
DARRIN KRUMWEIDE darrin@highendmag.com

Managing Editor
AMBER GODDARD amberg@dmgpublishing.com

Technical Editors
ALEX ALVAREZ alex@gnomon3d.com
DARRIN KRUMWEIDE darrin@gnomon3d.com

Art Director
Christina Wilkinson chris@highendmag.com

Design/Layout
Sabre Design artists@sabredesign.net

Highend Logo Design
Dmitry Savinoff dmitry@savinoff.com

HIGHEND MAGAZINE (ISSN 1541-5864) is published quarterly by DMG Publishing, 2756 North Green Valley Parkway #261, Henderson, Nevada 89014

ADVERTISING SALES sales@highendmag.com
CIRCULATION & SUBSCRIPTIONS subscriptions@highendmag.com

Application to mail at periodicals postage rates is pending at Henderson, Nevada, and additional mailing offices.

POSTMASTER:
Send address changes to subscriptions@highend.com or to Subscriptions, DMG Publishing, 2756 N Green Valley Pkwy #261, Henderson NV 89014

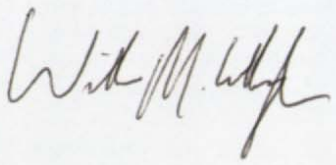
Will McCullough on the future of 3D.

The brainchild returns

WELL, WE MADE IT. ISSUE #2, HERE IT IS. Not as long of a wait as issue #1, and you'll notice some major differences this time around. We took some time to understand your suggestions from the first issue and made some adjustments. We listened and now we are proving that we are learning. This is a growing process for us, as well. Your feedback is essential to us and is helping us make this magazine what we all have been looking for. We have also started adding more features that you will see in the future, like support for Shake, that didn't quite make it into the first issue. You can expect to see other additions of features and supported software as the magazine grows. Just like highend3d, this is an evolution of what you, the reader and artist, request of us.

We are proud of where we've come from but are far from where we want to be. Keep the ideas, suggestions and art coming...we love it all. Pass the word around—Highend magazine is here to stay.

Take a look and enjoy the learning and the art; we hope you take from this issue ideas and inspiration that will help you to create.



CONTENTS

PAGE 40

FEATURES

DIGITAL DOUBLES
Convincing 3D replicates of existing actors, puppets, or elements used on set.
BY SEAN MILLS
4

DEMONSLAYER
Producing the Complex 3D Hero Spider Shot for the Latest Roger Corman Movie.
BY KAT CURRY, CRAIG SYLVESTER, KIRK SHINTANI AND PAM HOGARTH
40

KUNG POW
The arduous task of head replacement and Shake compositing techniques used on this unusual film.
BY JOHN LAFAUCE, JR.
46

ARTIST SPOTLIGHT

DIEGO AVESANI
49

PAGE 22

PAGE 46

TUTORIALS

XSI
Global Illumination rendering, without the wait
BY ED HARRISS
10

SHAKE
Generating a Matched Reveal Source in Shake
BY MATT PLEC
18

LEXUS' "'MINORITY REPORT' WAREHOUSE"
These effects greatly help to unify separately shot or created elements into a believable whole.
BY DARIUSH DERAKHSHANI
22

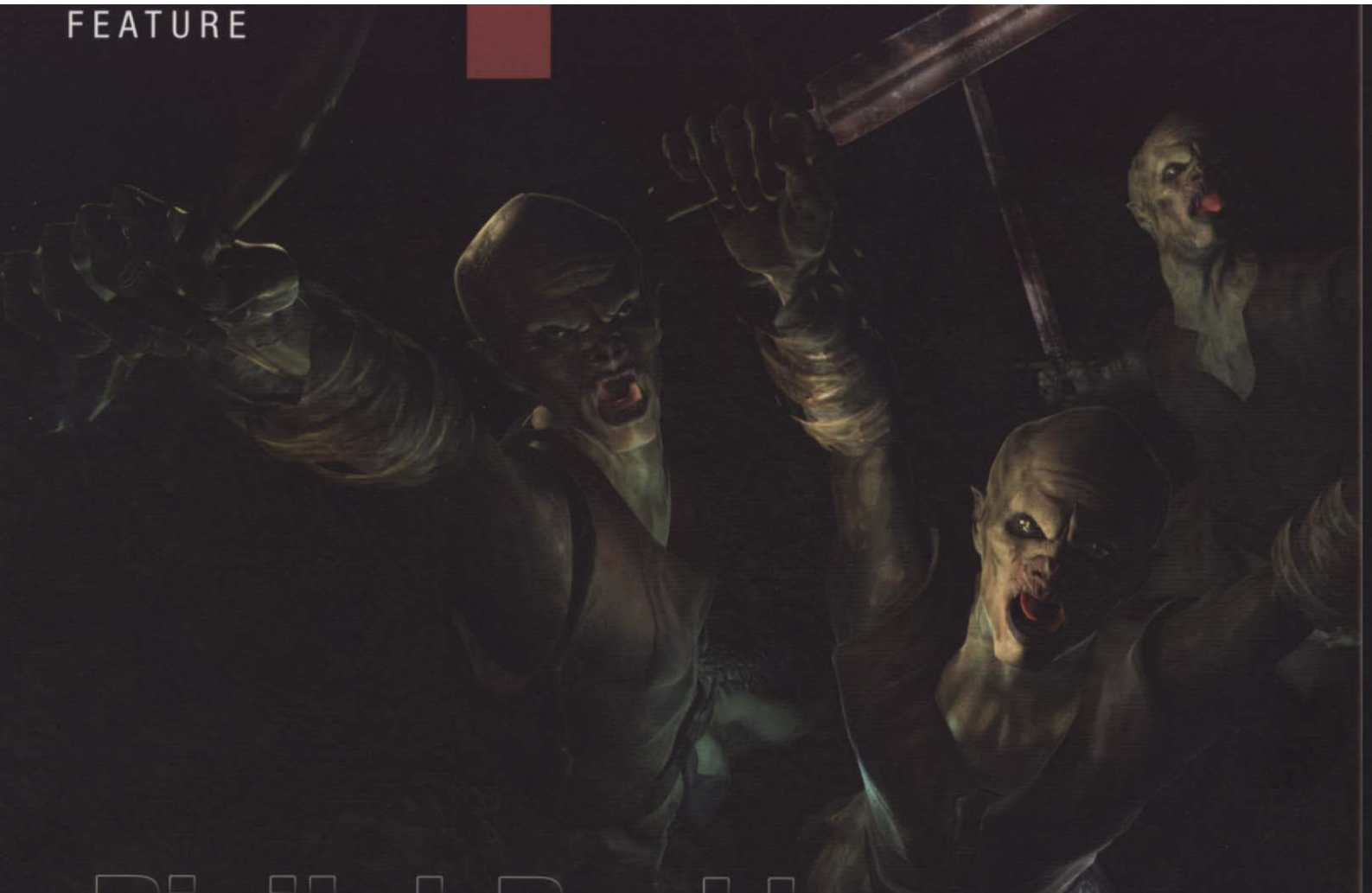
CREATING EFFECTS WITH CAMERA PROJECTION
BY DARRIN KRUMWEIDE, ASSOCIATE DIRECTOR, GNOMON INC.
34

GALLERY

26

COVER ART:
By Benoit le Longlée
alkos@3dvf.net

issue 2 • spring 2003

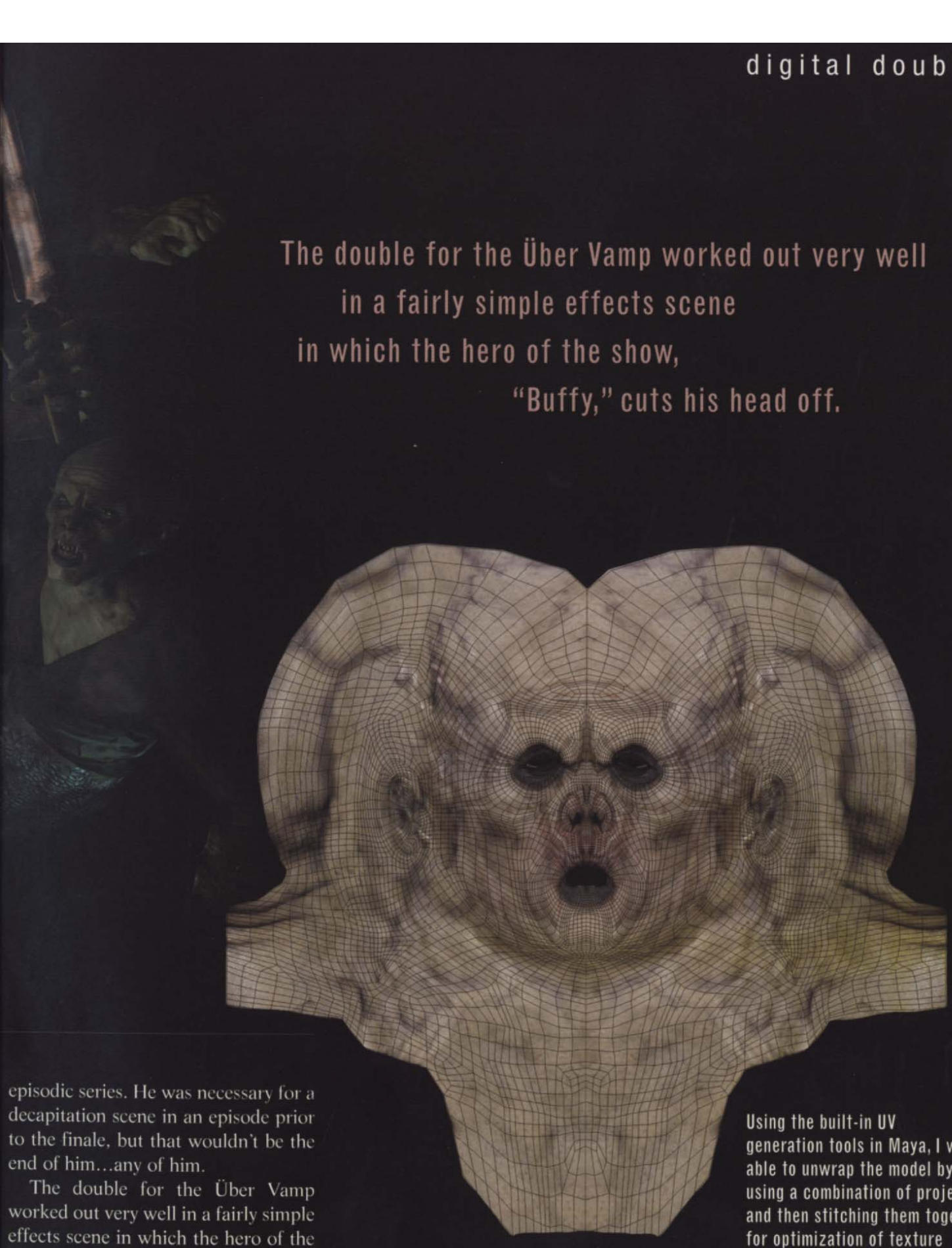


Digital Doubles for Visual Effects

BY SEAN MILLS

AS THE PUBLIC BECOMES more and more accustomed to seeing visual effects on broadcast television, audiences demand increasingly believable and impressive computer generated images. This means that though the production timeframe is as short as ever, animation studios and artists alike are constantly having the bar raised for quality. One of the current trends in visual effects is the use of "digital doubles," convincing 3D replicates of existing actors, puppets, or elements used on set. In the past, creating a convincing digital double could take months of research and development before actual production work could even begin. This article seeks to outline the workflow used at a studio like Zoic when the production calls for a believable digital double to be created within a limited timeframe.

With the recent advancements in production hardware and software the goal of photo-realistic digital doubles, or "synthesbians," can be achieved in a much more cost-effective timeframe. In this article, I will cover the steps and tools I used to create a digital double for the series finale of "Buffy the Vampire Slayer." This is the Über Vamp. Not just a vampire, but the UBER vampire, the originator of the vampire line in the



The double for the Über Vamp worked out very well in a fairly simple effects scene in which the hero of the show, "Buffy," cuts his head off.

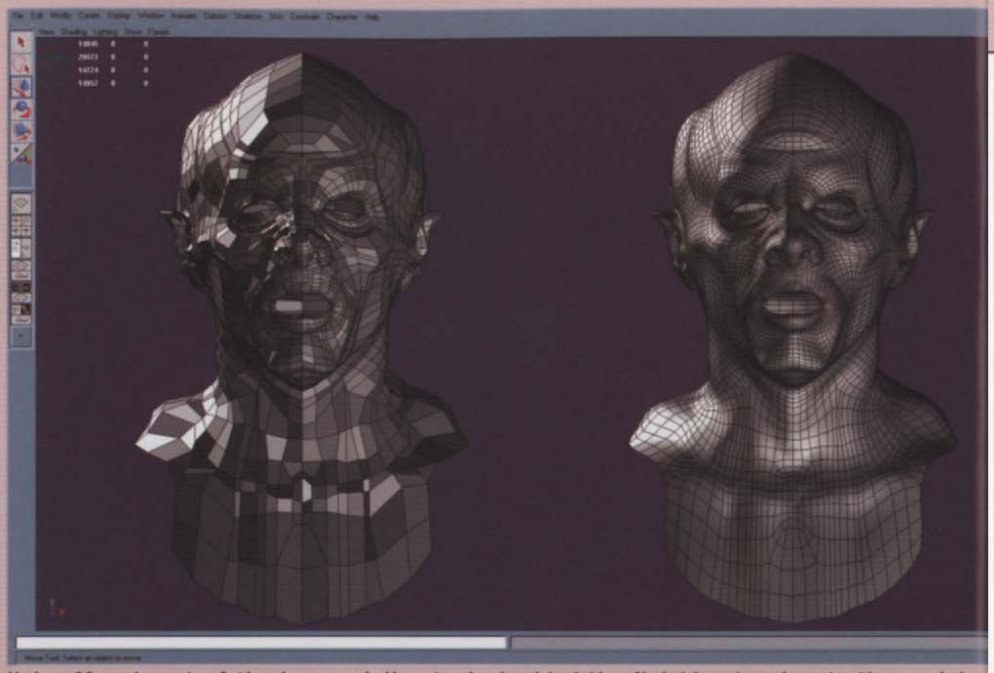
episodic series. He was necessary for a decapitation scene in an episode prior to the finale, but that wouldn't be the end of him...any of him.

The double for the Über Vamp worked out very well in a fairly simple effects scene in which the hero of the show, "Buffy," cuts his head off. So well in fact, Visual Effects supervisor and CG supervisor Andrew Orloff got the green-light for Zoic studios to begin production on a be-all end-all

Using the built-in UV generation tools in Maya, I was able to unwrap the model by using a combination of projections and then stitching them together for optimization of texture space and minimal streaking and stretching.

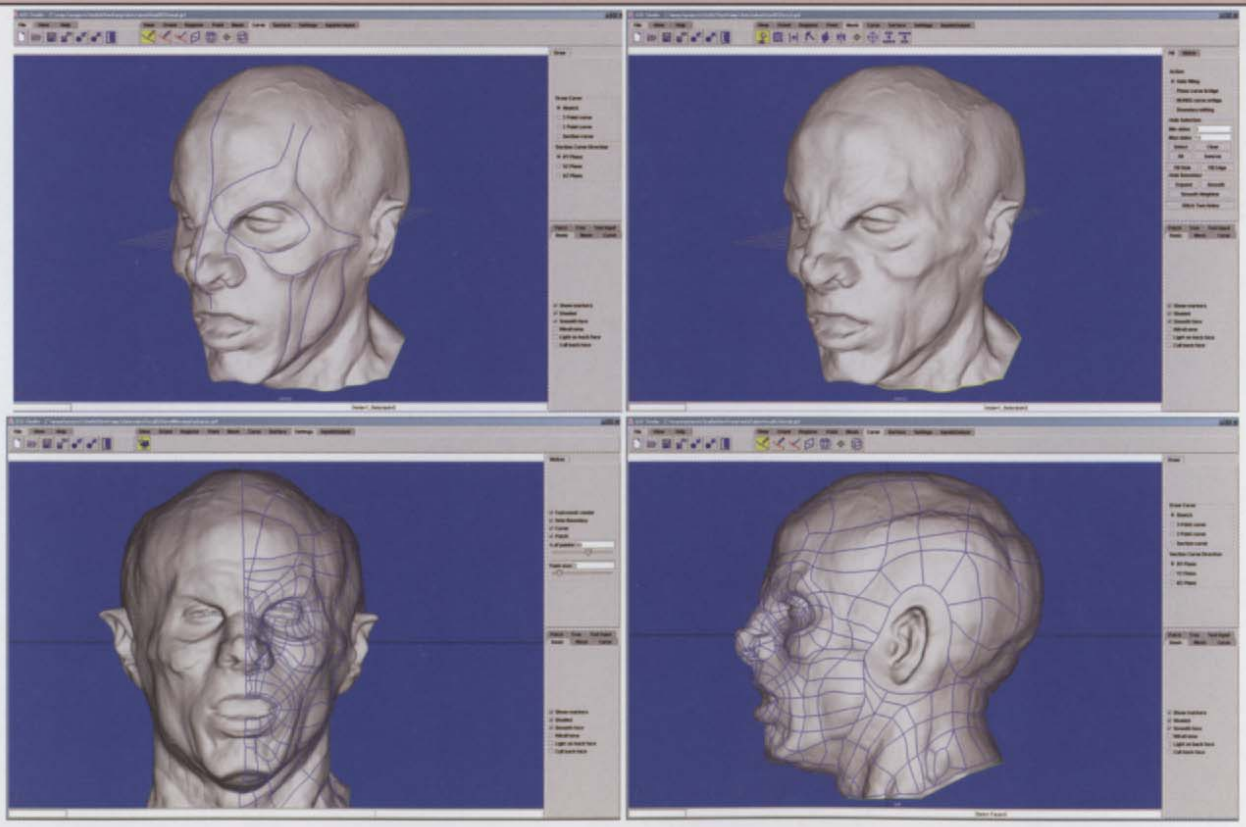
vampire hero shot in hell featuring thousands of our nocturnal villains.

The process of creating a digital double begins with scanning the talent, our live actor...the guy in the suit. To accomplish this, Zoic called on the mobile ability of Gentle Giant Studios, whose "scanner in a bus" arrived on set at a convenient time to the cast and crew of "Buffy." (We utilize a number of scanning service providers at Zoic Studios: Gentle Giant's mobile solution, Eyetronic's image-based scanning and Nexus Digital Studios full-service scanning department.) Inside was a small army of "Cyberware" scanning equipment. All that is required of the scanned subject is to hold still for about five seconds while the body is scanned. While an appropriate amount of data is collected for the body, areas of detail such as the



Using Maya's out-of-the-box modeling tools, I added the finishing touches to the model.

The Uber Vamp

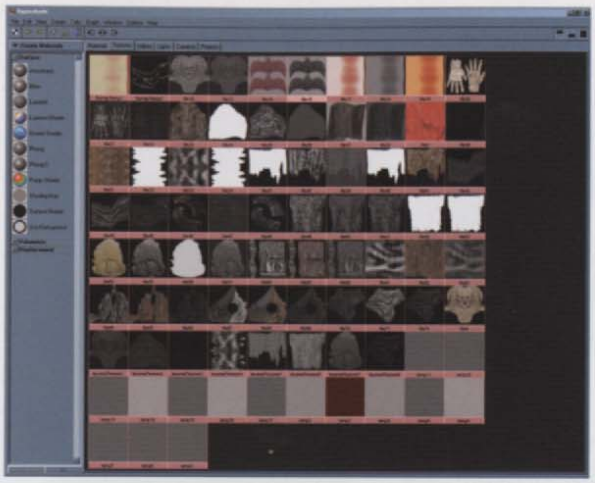


After cleaning up everything and having it sculpted and detailed to approximately what I want the final model to look like, I will begin using GSI Studios surfacing tools to layout a curve network across the model, focusing on facial structure and following muscle and major feature lines of the scan.

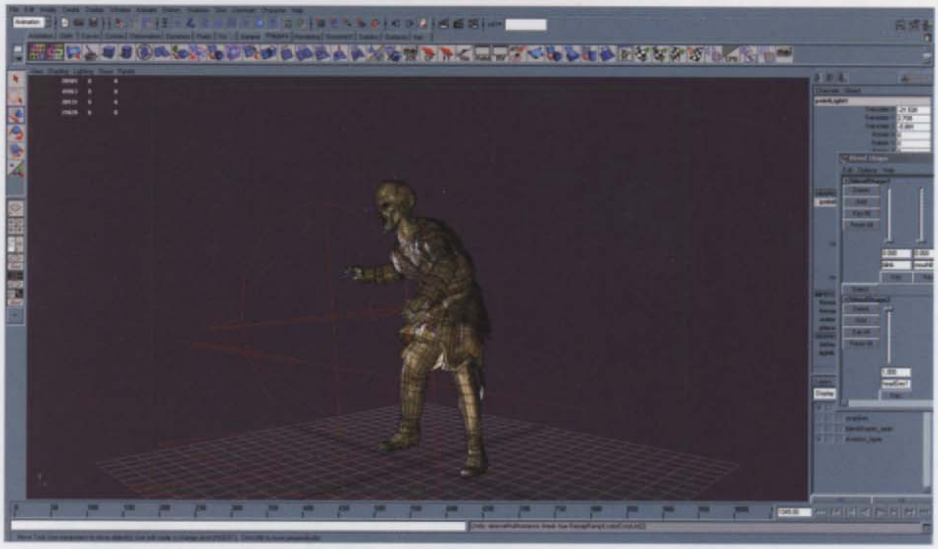
hands and face must be scanned separately using a higher resolution scanner. For these close-up scans the subject sits still as a ray of light is passed around the head cylindrically. These higher resolution scanners are also ideal for clay sculptures or miniatures. The result is a 3D geometric replication of our subject.

Once the data is delivered to the modelers, the real work begins. Because all 3D scanners use light to measure the varying depths of a surface, areas like the top of the head or behind the ears pose problems as the light is blocked, so the scanner cannot acquire any information from these areas. Using GSI Studios sculpting tools, I was able to model in by hand all the lost detail on the model using brush-based pushing and pulling techniques. In addition to the "problem areas" of 3D scanning, there is also a certain amount of "chatter" or "jaggies" across the model. This can sometimes be attributed to reflective or specular areas of the subject, and is to be expected with scan data. Ideally the subject should be as matte as possible...the less shiny and reflective areas the better. Unfortunately, that's not something that we really have much say on, so we work around it.

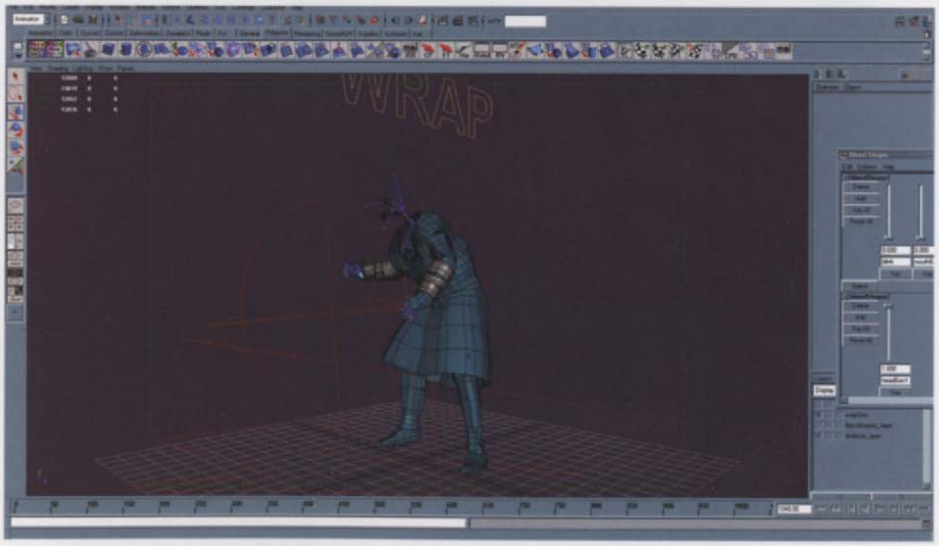
After cleaning up everything and having it sculpted and detailed to approximately what I want the final model to look like, I began using GSI Studios surfacing tools to layout a curve network across the model, focusing on facial structure and following muscle and major feature lines of the scan. Surfacing is necessary to create a usable production model. While scan data is polygonal geometry readable by most of today's animation packages, it's far too dense to do anything with. Surfacing allows us to create a usable model at a lower resolution (thousands of polygons vs. millions) model for production.



All together there were over 50 maps for the UberVamp including color, bump, spec and opacity maps for his skin and ragged clothing.

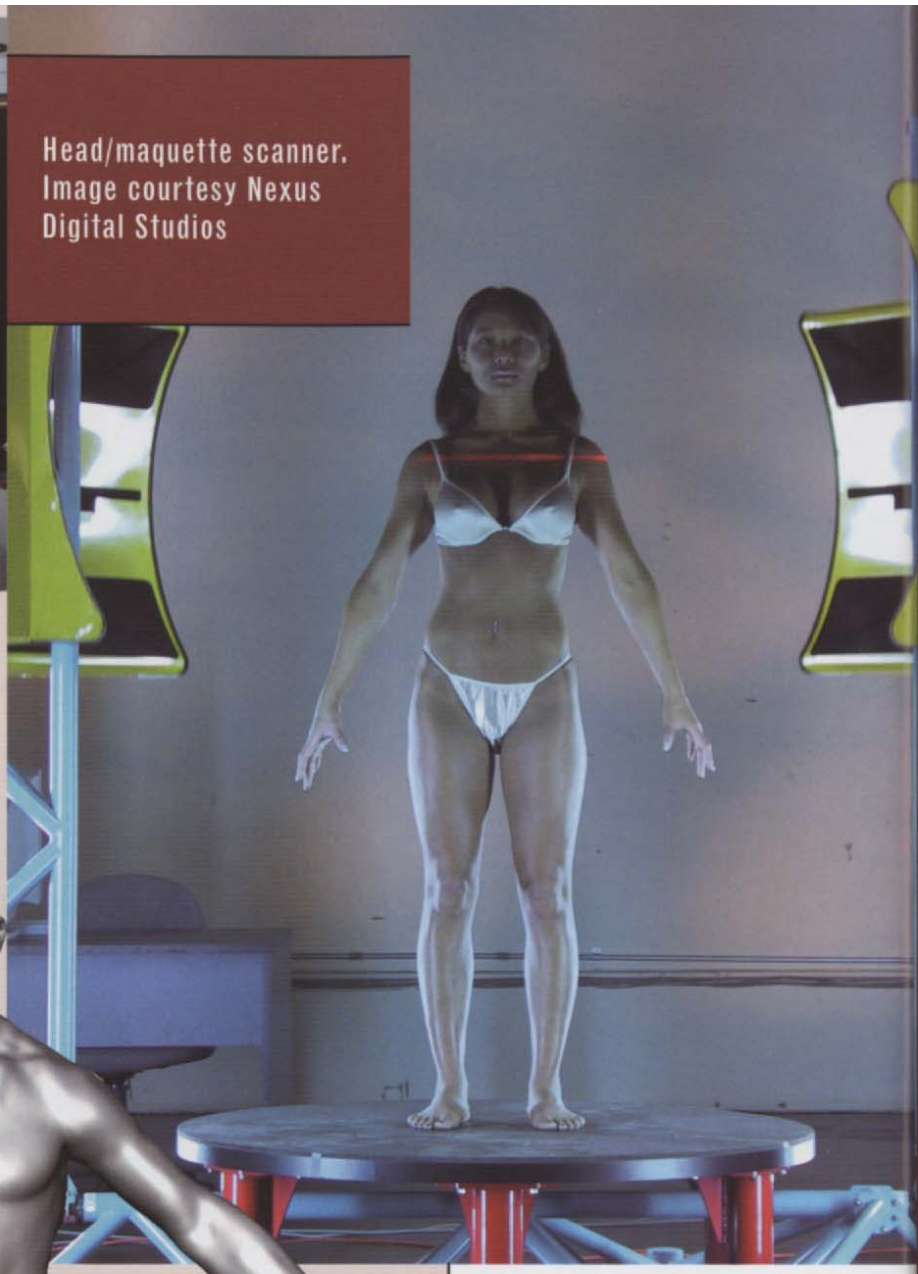


(mid-res geometry hooked up to mocap data)



(low-res wrap geometry hooked up to mocap data)

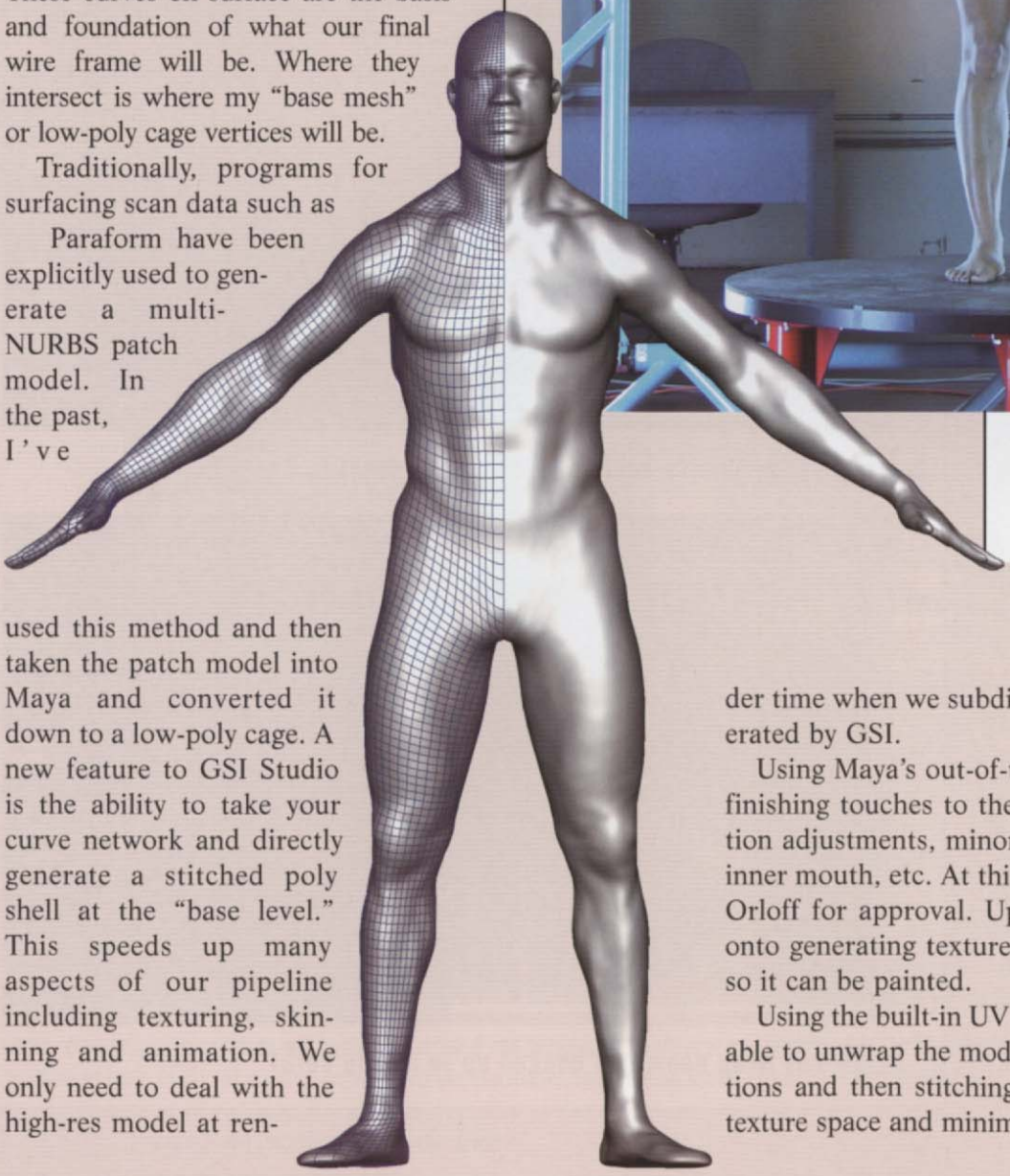
Head/maquette scanner. Image courtesy Nexus Digital Studios



Body scanner with subject. Image Courtesy Nexus Digital Studios

I used GSI Studio to interactively draw a network of curves directly on the scan data. Any needed changes or revisions could be made on the fly. These curves on surface are the basis and foundation of what our final wire frame will be. Where they intersect is where my "base mesh" or low-poly cage vertices will be.

Traditionally, programs for surfacing scan data such as Paraform have been explicitly used to generate a multi-NURBS patch model. In the past, I've used this method and then taken the patch model into Maya and converted it down to a low-poly cage. A new feature to GSI Studio is the ability to take your curve network and directly generate a stitched poly shell at the "base level." This speeds up many aspects of our pipeline including texturing, skinning and animation. We only need to deal with the high-res model at render time when we subdivide or smooth the base model generated by GSI.



Using Maya's out-of-the-box modeling tools, I added the finishing touches to the model, such as some local resolution adjustments, minor proportion changes, an additional inner mouth, etc. At this point, the model was submitted to Orloff for approval. Upon approval, it was time to move onto generating texture coordinates or UV's for the model so it can be painted.

Using the built-in UV generation tools in Maya, I was then able to unwrap the model by using a combination of projections and then stitching them together for optimization of texture space and minimal streaking and stretching.

Willow, of *Buffy the Vampire Slayer*, checks over her shoulder at effects work from Zoic Studios.

After laying out the UV's, the model is exported with these coordinates to Right Hemispheres Deep Paint 3D painting package. Using high-res photos taken on set of the actor, I began the texture maps. I was able to create a basis for the color map by projecting directly onto the surface from angles mimicking the angles the photos were taken at, then tweaking and painting the maps using Deep Paint over the course of a few days for the entire character. Once the base paint was laid I could then take these images with the wire frame laid over them into Photoshop, knowing exactly where the paint would affect the model.

After achieving a convincing color map, I used Photoshop and a pen tablet to derive and hand paint my bump and specular maps for the Uber Vamps head, body and clothing. Overall, there were over 50 maps for the ÜberVamp including color, bump, spec and opacity maps for his skin and ragged clothing.

Using these maps in conjunction with Maya's procedural layering and remapping texture nodes I was able to create a convincing shader network for all of the Über Vamp's parts. Once the shading and lighting were approved, the model was handed over to Orloff for skeleton construction and rigging, then to animator Jake Bergman to create a library of animations from a motion capture session held at Nexus Digital Studios. These animations were then mixed and incorporated into the shot accordingly.

Since the shot called for thousands of characters we needed to make L.O.D. models (level of detail), so the farther from the camera the character(s) are the less geometric detail there is. I went in and down-res'ed my base mesh cage and made several L.O.D versions. There was the primary base mesh for mid-shots, his smoothed big brother for close-up/hero shots, and versions descending from 20,000 polys all the way down to pre-rendered sprites for the smallest on screen characters. For skinning, I used Maya's wrap deformer to hold all the various ragged pieces of clothing and accessories together using one custom piece of geometry that animators could interact with in real-time. This one wrap deformer was able to deform all the various resolutions of the Über Vamp through one node, using Maya's file referencing we could easily choose which level to use on a per character basis. The head was directly bound to the skeleton and a bank of various "pirate blend shapes" were used for facial animation (he's not a speaking character, so there were no phonemes, but he did need to make various angry faces). Because the descending L.O.D.'s were derived from the original geometry, the UV's came through on each one so a saved weight map from the original head geometry was applied to the lower level head models.

The Über Vamp is just one of many digital doubles used for episodic effects work at Zoic Studios. The complete time from scan to skin including textures for this digital double was just over two full weeks. At the time of writing we are still in the final stages of lighting and shading for the "in hell" cave sequence, but all of the character work is signed off and going strong.

XSI Global Illumination Rendering, Without the Wait

BY ED HARRIS

I have not included a scene for this tutorial because I have found that it is easier for people to understand the concepts better if they use their own scene. This way they can concentrate on learning and less on the sample scene. So, my suggestion is for you to build a simple scene in which you can try out some of the techniques found in this article. All you really need are a few textured grids (to use for walls) and some primitives to go inside them.

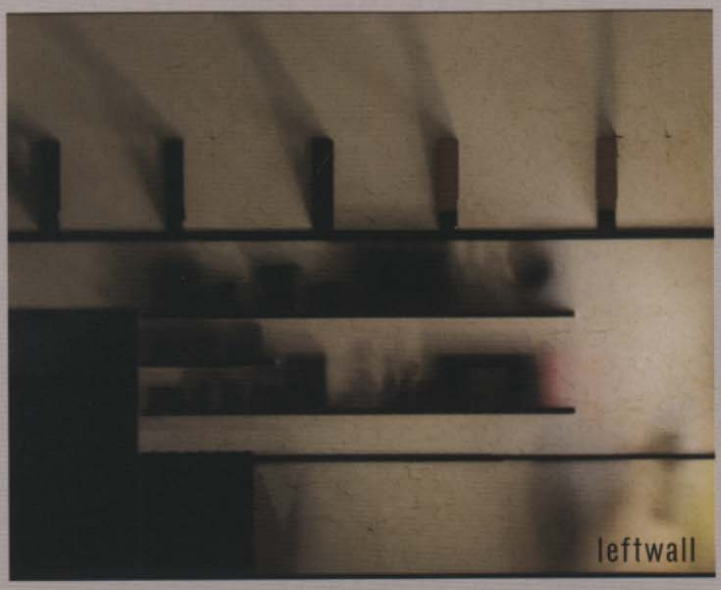
IF YOU'VE EVER USED GLOBAL ILLUMINATION (GI) or Final Gathering (FG), then you know that, depending on the scene complexity, it can add quite a bit to your rendering times. With the new optimizations that have been implemented in mental ray this is not as much of an issue as it used to be. But with scenes that have insane deadlines, (I'm sure you are all familiar with those) using GI and FG just might not be possible. If you usually light your scene traditionally, (I.E. without GI or FG) then you might not like the fact that effects like GI and FG take away some of the control that you are used to. However, using a combination of RenderMap techniques, compositing and HDR images, we can still utilize GI and FG in our scenes while retaining the control and low render times that are associated with a traditional lighting set up. Often this setup offers better results in less time than it would take to do it traditionally.

Take a look at these two images. They look like they were rendered with GI and FG turned on. But they weren't. They only look that way because of the specific texturing techniques used on them. In fact, this scene doesn't have any lights in it at all. As a result, it renders extremely fast. All of this is achieved by using textures that have been pre-lit using a function in XSI called RenderMap. >

Image 1

Image 2

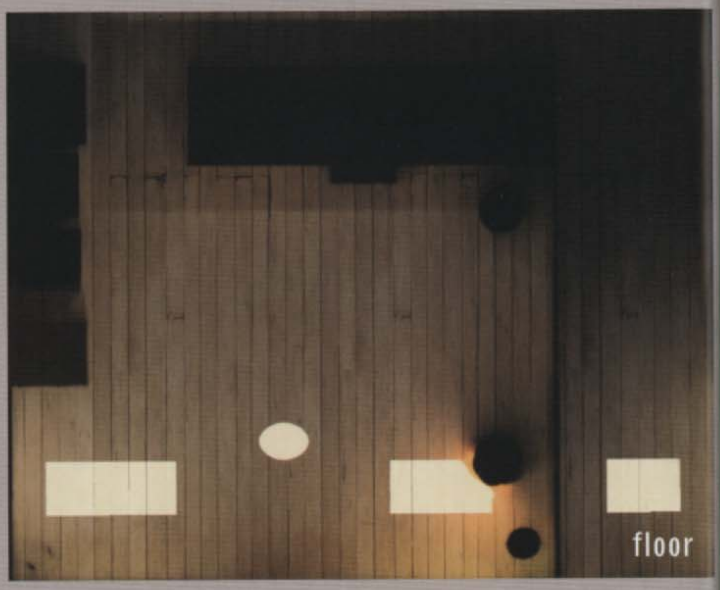
Floor Complete with Shadow



>If you are not familiar with RenderMap, it is an XSI tool that allows you to bake an object or objects texture, light, shadow and color vertice information into a single texture map. For example: Once the walls and floor in this scene have been rendermapped they will produce images that look like this:

LEFTWALL, RIGHTWALL, FLOOR

Notice how there are dark areas where objects are blocking the light and saturated areas where the light is bouncing off of brightly colored objects. Often rendermap is used on one object at a time. This may be a little time consuming, but it gives the user a lot of control over the rendermap generation. But, if you are short on time, it can be used on multiple objects simultaneously. This way you can rendermap your entire scene in one go. To find the rendermap tool, select an object and click **Get>Property>Rendermap**.



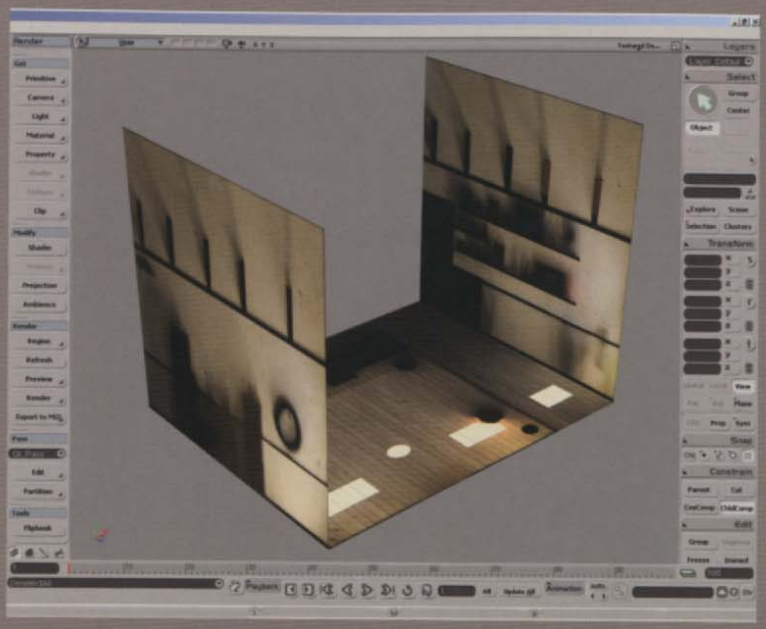
Once you've created your rendermaps, how do you use them to make your objects look like they are lit? If you just overlay your rendermaps on to objects with a standard material, like phong, blinn or strauss, then it will give you double shadows. What you need is to apply a constant shader to your objects. The constant shader is self-illuminating; this is what makes the rendermapped objects look like they have been lit. Once you have done that, you can turn off raytracing, all the lights, GI and FG. If you render your scene it will look exactly like it did before, but it will render in seconds instead of minutes (see image, *rendermap*).

Not only have you saved the performance hit that results from using GI and FG, but you'll also save any extra render time incurred by things such as: complex rendertrees, heavy area light usage, bump mapping, caustics, etc. Since all these effects are burned into the texture map, the renderer no longer has to calculate them. There is another huge benefit of using a technique like this. Since you've generated textures containing all of this information, you can tweak their look in a paint program or the FXTree.

You can paint light and shadows into areas of your rendermap to create effects that are difficult to achieve using standard lights, such as uneven shadow softening or coloration. Or you can use it to quickly fix mistakes that you've made. You can saturate portions of your scene while washing others out. You can paint extra detail into your caustics or break them up a little by running some filters on them. You can adjust all the rendermaps in your entire scene in the FXTree, making the lighting darker in one part of the scene or adding a green tint to another. Once you start using your imagination, you'll quickly discover that there really is no end to the tweaking

that you can do to your scene once you've rendermapped everything. Because there are no lights, it renders very fast. This makes it so easy to tweak your results in the render region that it is actually fun to work with.

Sounds too good to be true, doesn't it? This technique works great, especially if the only thing you've animated is the camera. But what do you do if one of your objects is animated? Remember, all of your objects shadows have been baked. When the objects move, the shadows they cast won't move with them. Additionally, constant shaded objects will not accept shadows cast by other objects. So, inserting new



rendermap

Rendermapping Tips

COURTESY OF DAVE LAJOIE

THE QUESTION:

Sometimes, when I render a scene with FG and then bake the result using rendermap, the FG doesn't look the same. Why?

THE ANSWER:

As you may know already, FG will only cache FG samples for visible surfaces. (visible from the camera) Since rendermap uses a "virtual camera" to render the surface to an image file, the FG requires a little extra handling to make it work properly.

Here is how that works:

Use the render region to build a FG map. This map will then be re-used in the render pass options during the rendermap process

a. In the render region options, enable FG, uncheck FG map Rebuild and give your FG map a brand new name. (To ensure that you are not using an old file.) This will allow mental ray to increase the FG map each time the camera is moved.

b. Draw a render region around the object, or objects, that you plan to rendermap. Orbit the camera around the object(s) so that it renders each of the surfaces at least once. This will accumulate FG data for your FG map.

c. Once the FG map accumulation process is done, turn off your render region

d. Go to the render options of the active pass, and set the FG map to point to the FG map file you just created. Make sure you turn off Rebuild since you don't want rendermap to erase it. The FG settings in the pass should be identical to the ones set in the region options. If they are not, it is possible MR will cast a few extra FG samples during the lightmap and rendermap process which would then be added to the FG map. I also suggest that you lock the FG map file if you plan to rendermap again later.

e. Start rendermap process and enjoy the show.

The method above is rather hard to script and automate since you need to use the camera to paint the FG map file and to accumulate the FG samples. The other approach is to increase the FG accuracy really, really, really high, like 6000, and force it to rebuild each time. The radius should be around 1 to 100, depending on the objects and its surroundings.

If you plan to use OpenGL to display your rendermap result, make sure you set the OpenGL image resolution to the rendermap resolution used. If you do not, some interpolation problems may occur and could cause black borders to show up around your image.

I hope this helps you to cook down really neat FG scenery and speed up rendering.

objects into the scene after rendermapping won't work either (Well, it will work. It just won't look right.). You might think that RenderMapping animated objects would require the use of an animated rendermap. This would work, but is probably a little too time consuming to be worthwhile. Fortunately there is a simple (and fast) solution to this problem.

The first step is to create the shadows cast by the animated object. In XSI this is very easy. Just use a shadow pass, (either by building your own pass or using one of the built-in passes) such as **Render>Pass>Edit>New Pass>Shadow**. Once you've done that, you'll need some lights to force your object to cast a shadow. You could just unhide the lights that you used when initially creating the image, but that might not give you satisfactory results because you've baked all of the lighting into the scene. You don't want to use GI and FG because that will defeat the purpose of rendermapping all of your surfaces in the first place. The easiest solution is to create a couple of new lights that give you a believable shadow, but for some environments, especially photoreal environments, you will need something a little more complex. Fortunately, XSI has an excellent tool for this problem. It is called a light rig.

The light rig tool will build a network of infinite lights whose color, direction and intensity are calculated by analyzing the pixels in an image. All of these lights are parented to a null so that you can rotate them, making it very easy to match your light rigs "direction" to the lighting in your scene.

HOW DOES THIS WORK? Imagine that you are working on a scene that consists of a blue floor, four red walls and a white ceiling with a white light in the center of it. Obviously, white light would come from above, blue light from below and red light from the sides. To use a light rig to light anything that is in this room, you would need an image with white at the top, red in the middle and blue at the bottom. The light rig tool will take that image and create some white lights, some blue lights and a lot of red lights. They will all have specific color values that are based on the pixel's color values found in that image.>

While the light rig tool sounds great, how are you going to create the image to drive it? You certainly can't paint it by hand, as that's too time consuming. What you need is a panoramic render of your entire scene, without the animated object in it. Again, XSI has a tool, this time it's the panoramic shader, that makes this very

Created for "20,000 Leagues: The Adventure Continues", this shot employs many of the tehniques discussed in this article. If you look through the windows of "Honest Abes Dive Shop" you can catch a glimpse of the scene on which this article is based.

easy to do. Here is how it works: Create a new camera, or use an existing one, and place it in the middle of your scene. Apply the panoramic shader to it. To do this, select your camera and click Modify>Shader. Then click **Lens Shaders>Add>Panoramic**. Render one frame using that camera. It will produce a render of your scene with a format similar to this (see image, *panorama*). Alternatively, you can rendermap a 100% reflective sphere instead of using the camera. Although this sometimes works slightly better, it is a little slower. Now that your panorama is rendered, all you have to do is click **Get>Light>Light Rig** From Image, choose your panorama render and click OK. This will generate all of the lights needed to render the perfect shadow pass for your rendermapped scene. Depending on the speed of your computer and the size of your image, light rig generation can take anywhere from a few seconds to a few minutes. Once you're done, all that is left to do is to press the render button and you'll end up with the perfect shadow matte (see image, *shadow pass*).

The next step is to integrate your animated object into the scene. Since it has not been rendermapped, you'll need to find a way to light it so that it looks like it fits into the scene. You could use the light rig that you created for the shadow pass, but to get accurate reflections you'll need a little more than that. The solution to this problem is to use XSI's Image Based Lighting pass. This pass will use the same panoramic image that you created earlier to generate reflections and lighting. To create an Image Based Lighting pass click **Render>Pass>Edit>New Pass>Image Based Lighting**. When prompted, click No, choose your image and click OK. The Image Based Lighting's Environment property page will open up. Change the Background value to 0. This will prevent the image from showing up behind your animated object.

Many times the decision between Light Rig and the Image Based Lighting pass comes down to personal preference.

Once you've created an Image Based Lighting Pass, you'll notice that the FG from the Image Based Lighting pass and the light the light rig will both be contributing to the scene. At this point you have to make a choice, since using both will wash out the lighting in your scene (see image, *Lightrig and Header*). To see the Image Based Lighting render alone then middle click the LightRig null and hide the rig **(Press the H key).** If you want to see the results from the Light Rig alone, leave it unhidden and turn off FG. Click **Render>Region>Options>Photon>Final Gathering** or **Render>Render>Options>Photon>Final Gathering**. Regardless of which option you choose, the reflections produced by the HDRI pass will work on your animated object.

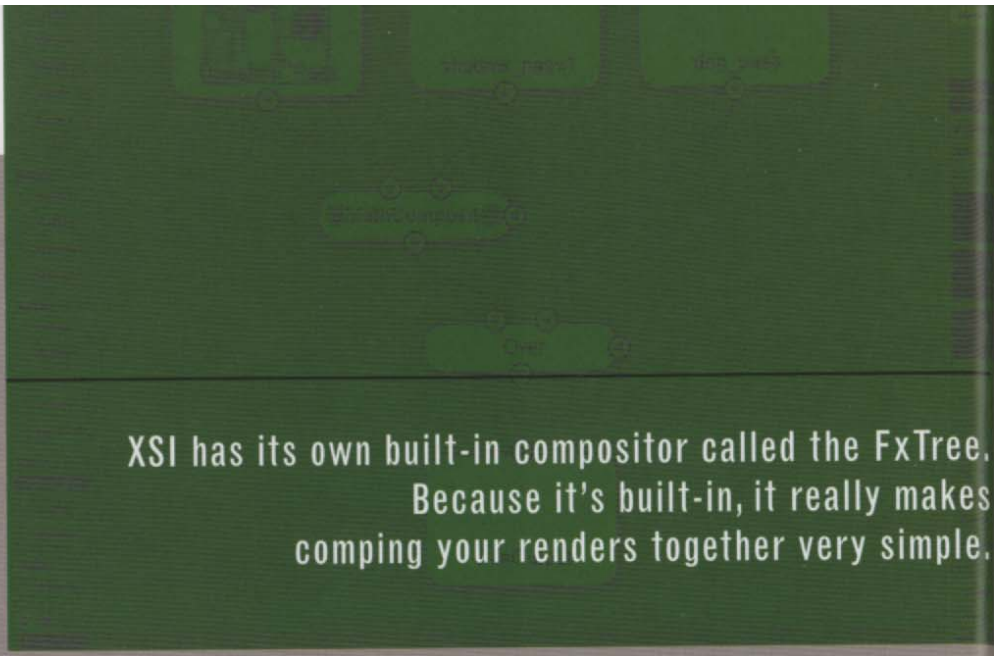
Many times the decision between Light Rig and the Image Based Lighting pass comes down to personal preference. Each render will look a little different; sometimes the Image Based Lighting pass looks better and sometimes the light rig image looks better (see images, *header render & lightrig*). You never know until you try. Sometimes a blend between the two images is best (see image, *header lightrig blend*). Depending on the complexity of your animated object, the render times when using Image Based Lighting may or may not be faster than when using the Light Rig. Even in this area it is a toss up.

Even with all these great tools, there are some situations where you have to improvise. If your animated object moves too>

close to a very bright object, then you might have to compensate with a stand-in object or light source. For example, the robot dog in this scene is very close to the direct sunlight hitting the floor. The way Light Rig and the Image Based Lighting Pass work, objects in the scene will be lit the same regardless of their location in space. This is because the light is generated at an infinite distance away from the objects in the scene.

The easiest way to solve this problem is to place a point light in the center of the affected area, in this case, the area where the direct sunlight is located. Another way to solve the problem is to create an object that is roughly the same shape and color as the affected area and assign it a constant material. (These three images, *floor light + header lightrig blend = dog floor light* should be used to make something like this: *light plus dog equals goodlight.*) This only works because constant materials will emit light when FG is turned on. While this will work, it is not the ideal solution, since the whole point of this exercise it to speed rendering up as much as possible. If you can get away with the point light solution, then you'll be much better off.

You've rendermapped your scene, created your shadows and lit your animated object. The only step left is to composite your rendering together to create the final image.

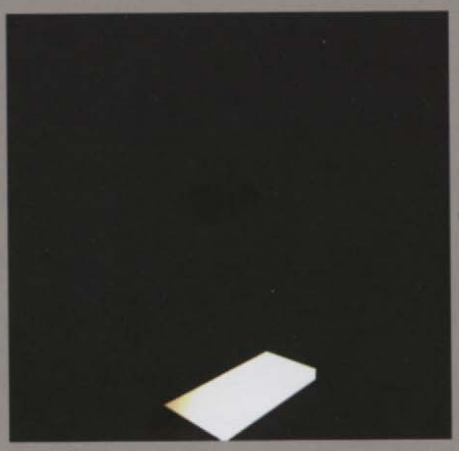


XSI has its own built-in compositor called the FxTree. Because it's built-in, it really makes comping your renders together very simple.

XSI has its own built-in compositor called the FxTree. Because it's built-in, it really makes comping your renders together very simple. Since you've rendered out your scene using passes, all you have to do is choose **FxTree>Passes>Get All Passes**. Every single one of your rendered image sequences will appear simultaneously in the FxTree workspace. All that is left now is to connect them together.

The first step is to composite the shadow on top of the background. Open up an FxTree and an FxViewer view port. Get a Math Composite node from **Ops>Composite>Math Composite**. Connect your Background Pass image to the left input of the Math Composite node and your Shadow Pass image to the right input. Double click the Math Composite node to bring up its property page. Choose Sub (A-B) from the Math Op drop down. This mode will take the white shadow matte and make it black (see image, *floorcomp with shadow* PAGE 2 OF STORY). For fun, you might want to connect a blur node between the Shadow Pass image and the Math Composite node (see image, *FxTree interface*).

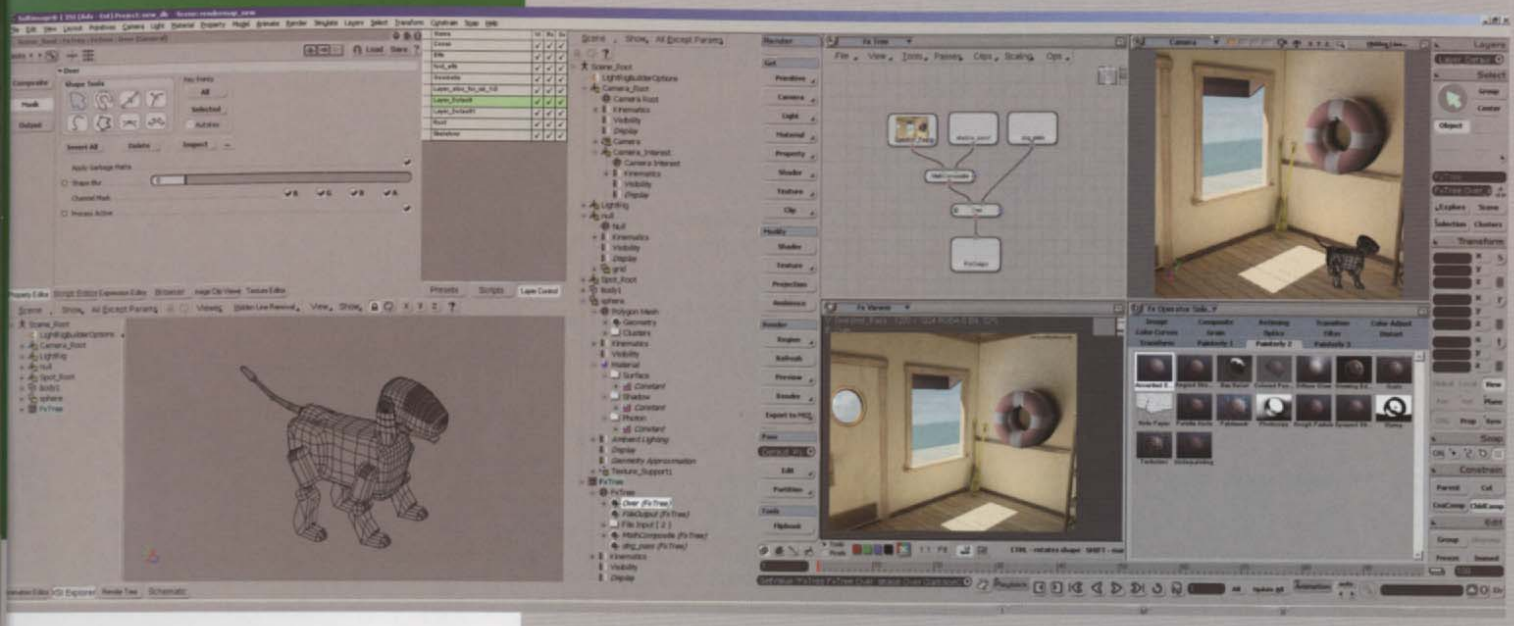
The second step is to composite your animated object on top of the background. Get an over node from **Ops>Composite>Over**. Connect the Math Composite node into the left input of the over node. Connect the animated character pass (HDRI pass) to the right connection of the over node. To view the results, move your mouse on top of the over node and click the blue circle. Now that your composite is built, all that is left is to render it



Floor Light

Dog Floor Light

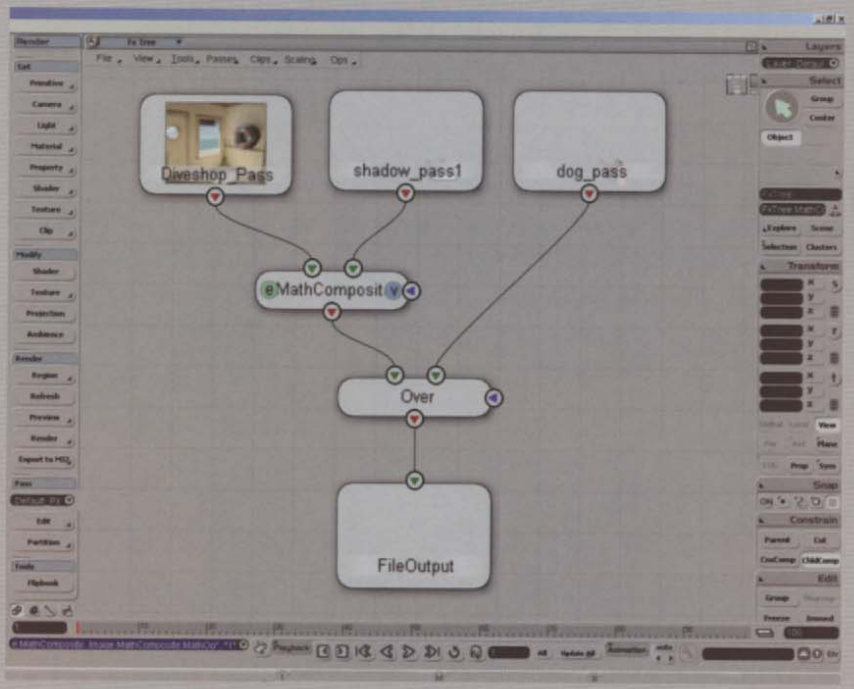
Light + dog = goodlight



FxTree interface

to disk. Get a File Output node from **Ops>Image>File Output**. Connect the over node to the File Output node. When you're finished, your tree should look something like *FxTree Composition*.

Double click the File Output node to open its property page. Set the file destination, name and type and then click the render button. That's it! You've created your first scanline rendered scene that looks exactly like it was lit using GI and FG.



FxTree Composition

CONCLUSION

It might take you a project or two to become familiar with the techniques described in this article. During this period it could take you a little longer to get those projects done. But, once you have gotten over the initial learning curve, you will be able to get more work done in less time, your work will look better and render faster. In fact, this technique works so well that it is used in just about every area of computer graphics in the industry, from videogames to blockbuster movies. Did you know, that the battleships and the airplanes in "Pearl Harbor" were lit using a technique almost identical to this one? In fact, it worked so well that even the director couldn't tell the difference between the CG planes and the real ones.

"Did you know, that the battleships and the airplanes in **Pearl Harbor** were lit using a technique almost identical to this one?"

Generating a Matched Reveal Source in Shake

BY MATT PLEC

Although Shake is best known as a compositing tool, a few versions back a vector paint node was added. While you won't be throwing out Photoshop any time soon, it does provide the ability to do a lot of those quick touch-up fixes that seem to be a part of just about every shot. And best of all, you won't have to file out to 8-bit and go work in another application; you can stay at 16 or even floating point straight through.

The QuickPaint node has only the most fundamental tools, as of version 2.51, but thanks to Shake's flexibility you can get quite a lot out of it. One of the primary shortcomings of QuickPaint is that the Reveal mode offers no control over the position and time offset of the input clip used as a source. You won't be able to change that for each stroke within a frame — that's a shortcoming we'll need to wait for the Shake developers to address — but with judicious use of expressions and in combination with other Shake nodes, it's possible to set up a paint-through source that can easily be repositioned and time-shifted for each frame requiring touch up.

In the shot that I'll be working through here, from a short "Star Wars" fan-film about the invention of the first light saber, a sputtering beam element is to be added to the practical light saber handle in the actor's hands. Unfortunately, during the shoot no one realized that the wire sticking out of the handle would be a problem—it's going to be covered by the light saber beam, right? Well, not always. When it's sputtering on everything's great. But when it sputters off, we've got a problem.

A typical in-the-comp approach to this situation is to create a clean plate, track it into place, add grain as needed, and then use a RotoShape to replace the problem area in the original footage. Unfortunately, this shot has the added complication of moving camera creating a change in per spective over the duration of the sho (Image Filmstrip).

Even after matchmoving the clean plat into position, that perspective chang means that much of the shot still won look quite right. What's really needed is clean plate that corresponds to each of th original frames. In light of that, I'll try a di ferent approach, matchmoving a fram near the current one into place for use a a paint source. I won't have a complet clean frame, but with all the movement the shot I can slip the footage a few frame so that I get enough clean background any particular time to paint out the wi

and the perspective will be nearly the same. A nice consequence of this approach is that as long as the source frame is kept changing I get the grain for free. The trick is in setting up so that whichever frame I want to use at any given time is in the correct position.

Track the Plate

Since I'm going to be repositioning one image to match another, I know I'll need to get tracking information on the plate. A full tracking tutorial is beyond the scope of this article, but there are a few aspects of this shot worth noting. (The Shake documentation provides a thorough (and entertaining) discussion of the subject under Overview / Tracking and in the Tutorials / Advanced / Tracking sections.) A look at the footage for this section of the shot shows that the wire to be removed is generally passing over the window in back (Image OrigFootage.

Filmstrip

Given the extent of the moving camera throughout the shot, a two-point match-move would not account for all the changes in the appearance of the window area. A four-point track will allow me to pin edges of the window in place to get the most accurate match possible in that window area. This may cause the image to stretch and match poorly farther out towards the edge of frame, but only the areas within and near the window are required for the paint-through. As long as that lines up well the rest doesn't matter; it won't be used.

The tracker's default (and most reliable) mode is luminance. Here I've attached a Reorder (to "lll," putting luminance in each channel) and then an Expand to guide me as I position the Tracker's feature areas (Image TrackPlacement).

Sometimes it's a bit hard to tell what might be good track points; our eyes and brain are not processing the image in the same way that the Tracker's logic will. We may recognize certain groups of pixels as unique features even though they don't look so well-defined to Shake. Before settling on tracking points it can be helpful to take a look at the image's luminance and

Original Footage

Track Placement

slide the range around a bit to see what pops out (Image ReorderExpand).

While it is sometimes helpful to preprocess the image a bit before tracking, particularly to remove grain that might throw off the correlation computations, I'll first try tracking directly on the original source footage. The Tracker is built to expect regular images and does some normalization itself to best compute the correlation. If it seems to be having trouble, I'll spend some time adjusting the image to help it out. With this footage it doesn't turn out to be necessary.

In the simplest case, this tracking data would now be used to lock another piece of footage to the original; for example, some prepared footage placed into an on-set television screen. Another common situation involves a piece of footage that's moving and needs to be tracked into position on the original moving plate. This shot is a variation on that second case, but since the second piece of moving footage is the same as the first, I've already got the tracking data. The trick is setting up so that I can line up any arbitrary frame of it with any other, rather than just lining up one running sequence frame by frame with the other.

Re-time the Reveal Source

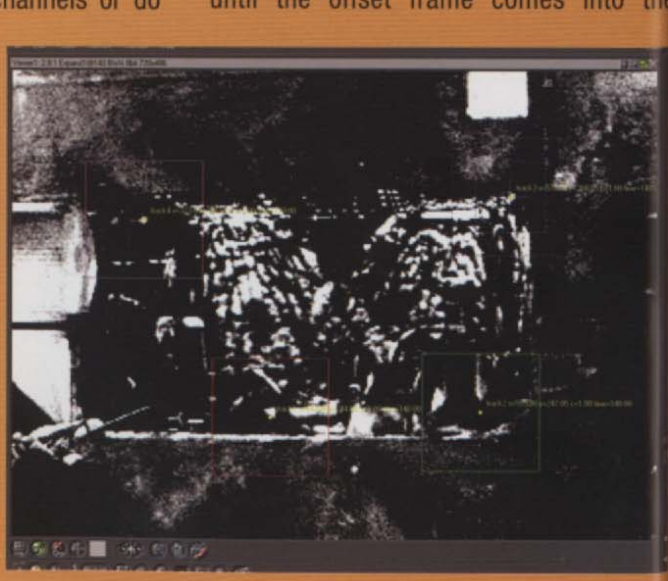
First off, I need to get control of the reveal source's timing. Since I'm going to need the original plate as well as the re-timed version, I'll duplicate the original FileIn and connect a TimeX to the copy, naming it "sourcetime." I chose not to connect the TimeX to the original FileIn because sending a node's output to differently timed branches can generate unpredictable results in Shake. By default, TimeX's newTime parameter follows the global time. I'd rather choose the new time as an offset from the current one, so I'll create a local variable, called offset, for this. I'll make it a float type variable just to be consistent with Shake's global time, even though I'm not working on fields for this shot so it's not strictly necessary. Now I can set the newTime parameter to "time - offset" and use the offset parameter to easily specify any number of frames backwards or forwards necessary to get a section of image suitable for painting through over the wire (Image TimeX).

Position the Reveal Source

To get this re-timed source into position, I'll start with a standard MatchMove node, trackType set to 4 point, and the four track points linked to the corresponding track data parameters in the original Tracker node. I could have tracked directly in the MatchMove node, but it's limited to four channels of track data. To give myself as much flexibility as possible I generally do the tracking in a Tracker node where I can add as many track data channels as I want. This gives me the ability to average several of the track data channels or do any kind of preprocessing or manipulation needed to create four usable points.

Ordinarily, I'd now simply copy the values from the first frame of track data to the source X/Y position parameters. This would give me that first fram lined up nicely with whatever frame of th original I happen to be working or However, I need the lineup to be defined b whichever frame is coming into th MatchMove's foreground, not by one fixe frame. I'll get this by setting an expressio on the source position parameters that ref erences the track data values at the sam time as the input frame, which is define by the value of newTime in my TimeX nod "sourcetime." For example, I se source1XPosition to track1X@@(source time.newTime). Notice the "@@" notation which tells Shake to evaluate the parame ter name preceding it at the time define by the expression following it. The paren theses aren't essential here, but use o them is a good habit to get into to avoi unexpected results when using a mor complex expression (Image MatchMove).

I can A/B the MatchMove against the orig inal footage to check the lineup. (Don't for get to set the MatchMove's applyTransform active and outputType to Foreground.) B aware that if your offset is set to tak frames from before the current fram you'll see a mismatch on initial frame until the offset frame comes into th



Reorder Expand

TimeX

MatchMove

tracked data. A similar situation occurs at the end if your offset source is after the current frame. Before and after the keyed track data, values will be an extrapolation of the existing track data. Fortunately this is easy enough to avoid by starting out with your offset keyed to frames ahead, then move to frames behind. This whole setup is designed specifically to make it possible to use any frame in the sequence as a source—take advantage of it! Key the offset to the closest frame that provides a clear area to reveal. A Step curve is a good choice for the offset keys as it insures that you'll get only the offsets you've keyed, never an interpolation of them.

Apply the Reveal Source

You might notice that the transformation introduces a bit of softness in the match-moved images relative to the original source. I added a Sharpen to counter this and used the A/B view again to adjust parameters until the images matched acceptably.

Now I'm finally ready to do the real work of the shot. I'll add a QuickPaint with its first input connected to the original source footage and its second input to the replacement I've generated (Image QuickPaint).

I load the Viewer's A buffer with the QuickPaint node and its B buffer with the reveal source so I can quickly toggle back and forth to see if the area I'm going to reveal is clear of the wire in the reveal source. I've also got the TimeX node loaded into the Parameters 2 tab so that I can quickly switch to a different reveal source frame. Just keep in mind that you can only choose one frame for any particular global time. In a more complicated shot it might be necessary to generate clean areas at different source times to fix other parts of the frame. This can be done by duplicating the reveal source branch and applying the output of this copy through another QuickPaint node.

Quickpaint

Other Applications

Keep in mind that this technique isn't limited to use with QuickPaint. I could have just as easily used a RotoShape node to create a cutout for layering into the original. In this case, the wire is small enough that I'm able to simply paint it out without noticeable boiling. However, for removing a bigger object, where boiling is more likely to be noticeable, a RotoShape would generate a more consistently defined reveal section. Likewise, if a key could be gotten on the object to be removed, that could be used to define the area of the reveal source to be merged into the original footage. This re-time and match procedure is general purpose, enabling you to quickly set up and easily manage matching an arbitrary frame from a sequence to any other frame in that sequence. Even if QuickPaint evolves to support manipulation of the reveal source's position, this technique provides access to that functionality in combination with any of the tools available in Shake.

Lexus' "Minority Report Warehouse"

BY DARIUSH DERAKHSHANI

COMPUTER GENERATED IMAGERY needn't always bash you over the head in a scene. Some of the best CG created is supposed to be hardly noticeable in the shot, and is only there to support the credibility of the live action. It's sad, but true, that the better the work we do, the more unnoticed it goes. Creating environmental effects is the biggest example. Ranging from set enhancements to weather effects, environment CG plays a significant supportive role in more shots than you think, and that's the point.

It is practically a norm now that scenes are shot with the knowledge that different elements will be composed together and CGI will be added to augment these scenes' reality or increase their drama, all well after the film has rolled and the location is left behind. Weather is changed, skies altered, terrains adjusted, crowds created, buildings made, all to cohere with the live action elements of the shoot into a convincing setting through which the tale is told. These effects greatly help to unify separately shot or created elements into a believable whole.

That cohesion rests solely on the plausibility of all the composed elements and their interaction with each other. It's one thing to create a dramatic atmosphere with CG to add into a live action scene, but it is ultimately the goal to make the viewer unaware of that addition; to make them believe everything in that scene is in unity and has always been so.

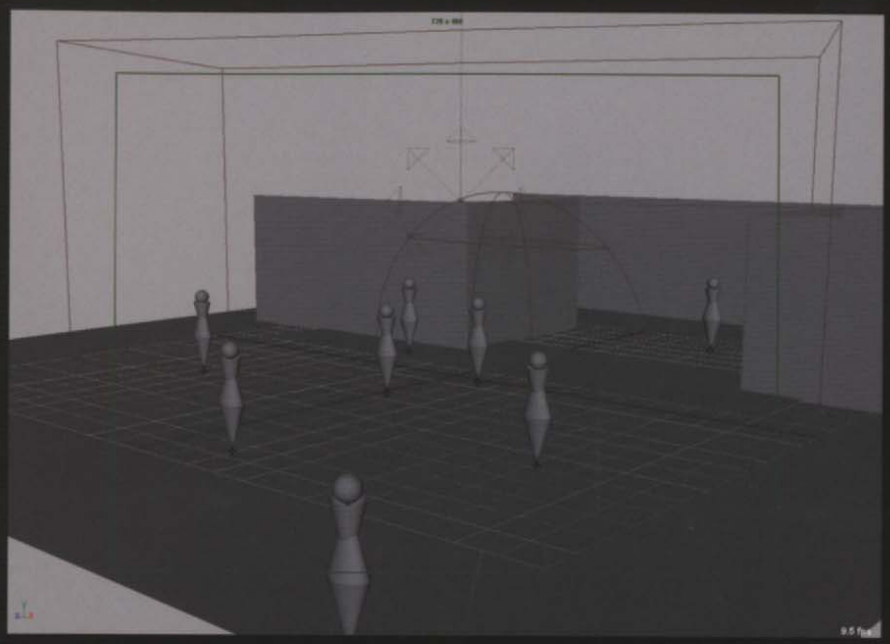
The exertion in creating these environment effects lies with their need to look photo-real and animate according to the actual and fictional physics of the scene. Getting them to blend in may be one thing, but getting them to look as if they belong in that space can be quite another.

With Lexus' commercial " 'Minority Report' Warehouse," some of our CG staff at Sight Effects of Venice, Ca. was charged with the task of creating set extensions and environment effects to place the characters of the spot into the same style futuristic city laid out in the Steven Spielberg film "Minority Report."

As the spot's newly designed Lexus sits hidden inside a warehouse, three jet-pack police fly into the scene and send in their hovering bots to locate the car. When the bots locate it, a chase ensues; our hero in the Lexus outsmarts the flying meanies and drives off into the nighttime cityscape with the bad guys struggling behind to keep in pursuit.

Shot on location in an industrial warehouse district, the environmental CG work encompassed creating set exten-

FIGURE SETUP 1



sions and backdrops for the location shoot, as well as layers of steam and fog mixed into almost all the shots. The set extensions, of course, created the fictional futuristic backgrounds, but the environments were called upon to add a dimension of reality to the digital mattes and composites, as well as help everything come together.

First, a hazy fog needed to be created and added to all the external shots of the spot, even those with some real fog in them. Maya hardware-rendered particles were used for this effect. However, creating a simple fog layer in Maya and compositing it over a live action plate will flatten out the image. Using just a single fog layer, no matter with what detailed depth it was rendered, will look superimposed on top of the shot in a composite.

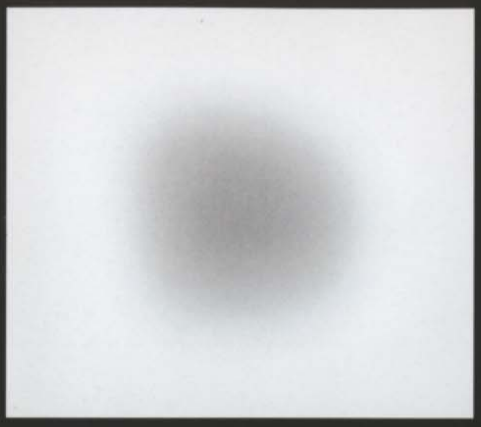
Add to that characters moving around in the scene, and the flatness of that fog layer is accentuated even more since the characters have no effect on the CG fog layer rendered and comped in front of them. It will stick out like a glazed ham at a vegetarian dinner to see an actor move around supposedly in a fog, but not be able to see the fog react to his movements. For this spot a good amount of our time was spent in making several depths of fog layers for each shot, most of which were interactive with the characters that moved in and out of the fog.

Creating an Environmental Fog Effect

The solution is to create multiple layers of fog that are then composited behind, with, and in front of the characters. By slicing the scene into layers of depth, you can map out quadrants of depth for your fog effects. The layers to be composited in the same plane as the characters can then be animated to be interactive with their movements and composited appropriately. Once all the layers are integrated with the live action plate, the effect is seamless and becomes part of the original scene.

The first step is to load the background images into Maya as image planes. That will give you a sense of scale, and allow you to see any movement within the scene. Build out proxies for most if not all objects in the scene, specifically the ones that may block parts of the environment, such as walls, buildings, characters and such. These will act as masks to allow the fog to have depth where appropriate and no depth when not within any layer. The CG proxy objects should have flat black shading on them and the CG edges must match the live action edges to make sure the fog appears and disappears behind the live action counterparts properly.

Now you can render your layers of moving fog and use a compositor to matte out the live action people and elements to lay in fog elements behind occluding objects. The 3D mask objects you'll be using within the scene would be used to mask areas within a certain layer. If you define your layers well enough, you can minimize the amount of 3D tracking by placing zones between as many of the live action elements of the scene as possible. Only when an occluding object falls inside a layer would you really need to mask the areas with a flat black 3D object as above.



SPRITE EXAMPLE

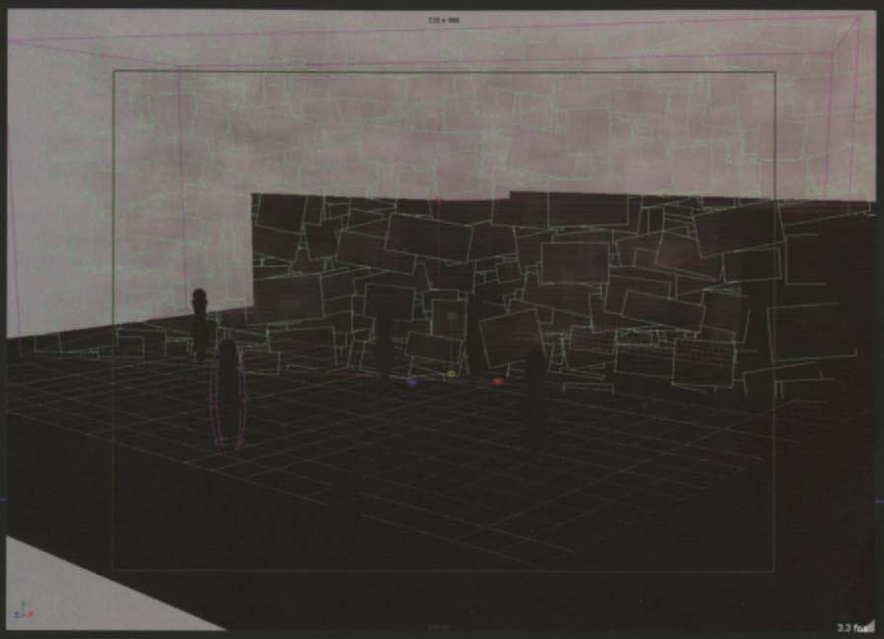


FIGURE SETUP 2

This is a good time to make a rough floor plan of the scene to define layers of depth. Track the camera and proxy objects to anything that moves around in the scene enough to disrupt the fog around them. You'll want to track the gross movement of people, cars, hover bots, etc. so you can make them reactive to the particles in the scene.

Next, create a volume emitter and fit it to the scene's background. It's generally better to scale the volume emitter to a good depth, to get variation in the movement of the particles. Position the emitter for the background and scale it accordingly to fill the background. FIGURE SETUP 1 shows a simple floor plan and emitter placed for the background fog. Simple proxies are positioned and tracked to the live action elements.

The emitter will emit sprites with a very faint puff assigned to them (see SPRITE EXAMPLE). Make sure the sprite image doesn't break the raster, meaning the puff shouldn't reach the edge of its frame. You wouldn't want sharp edges in your sprites showing up like panty lines at a prom.

The OpacityPP needs to be set ridiculously low for each particle (with a random expression to vary the transparency slightly between particles) so that only a good massing of particles will show up. Sure, it's not terribly efficient to have tons of particles, but it looks best this way. Scale the sprites to look as if they are in the distant background. You'll want per particle scale attributes so you can set expressions to randomize their sizes a bit. Create them wider than their height at about a 2:1 ratio to give them horizontality using their spriteScaleXPP and spriteScaleYPP. You don't want them too big to appear large and blown out, but you definitely don't want them too small. That would make them appear spotty, like far too many smoky effects can.

Add and randomize a spriteTwistPP attribute as a creation expression so they have slightly different horizontal orientations. This will help make the distant fog look deep and layered from the ground up. You can even add a very slight rocking back and forth using a runtime spriteTwistPP expression.

Lastly, set the lifespan mode to random range with 3.0 lifespan and abou 1.0 random. You could also put lifespanPP on them, but this will work jus as well. Based on the scaling in my scen (yours will be different), here are the cre ation expressions to set the size, rotatio and opacity:

```
particleShape1.opacityPP = rand
(0.01,0.06);
particleShape1.spriteScaleYPP = rand
(0.5,1.5);
particleShape1.spriteScaleXPP = rand
(1,2.5);
particleShape1.spriteTwistPP = rand
(-15,15)
```

Now to set the emitter attributes You'll want to create a subtle backgroun layer, so set the emission rate to around 50 to 650. You'll need to adjust this accordin to the opacity and size of your sprites, bu remember subtlety is the rule here.

Part of what will truly sell this distan layer is its motion. Since it is for the back ground, behind all the other action, you'

want a smidge of noticeable movement, but not enough to be distracting. Set the emitters speeds to about -.25 along the axis, to make the fog travel down slightly, away from center at about .2, around axis at 0, a random direction of about .5, and a directional speed of 1.0 in a lateral movement. For example, set it to travel slightly down and to the right of frame at a slow rate (see FIGURE SETUP 2).

The hardware render buffer and its attributes is where you'll win or lose this fight. The intent is to have this be a very faint rear fog layer, so you should use a high number of render passes in Multi-Pass Rendering with a ridiculously high motion blur setting, such as 36 passes with 25 blur (just make sure your particles are moving slow enough). Render your sequence and comp it in the scene, carefully controlling color and alpha settings to mix it back to be barely noticeable, but definitely there.

Now why use particles here way in the back? Why not use a volume fog with an animated 3D texture applied for its density? Simply put, we'll be using the same particle set up for all the other layers of this fog. We'll want there to be continuity, even if it is barely noticeable. Plus, if there is a change in the scene, or an object or character moves into the background, you will have to make this fog layer reactive to not only their motion, but the air currents their movement causes. Fog and smoke move according to the air and temperature currents that carry them.

After the background is created, you can use the same scene file as a template for all the other layers. All that will change will be scaling of the emitter and sprites, movement speeds and perhaps directions, and perhaps a little opacity change, generally lower, since several layers will be added depending on the depth of the scene.

You'll want to keep the general movement the same in all the layers, with a little variation, so adjust the emitter speeds and any runtime movement to the sprite twists. The major difference here in these foreground layers is interaction. This will really help sell the shot.

Track volume radial fields to the character or object motions in the live action plate, and set them to repel the sprites as they get near them. Using a relatively high magnitude will make the particles fly away too fast, but setting it too low may not make them move fast enough when an object moves around. A radial magnitude of 4 or 5 should be fine. The volume shape of the field will give you a visual idea of where the field and particles will contact, making it easier to track to live action.

Rendering all the particle layers will give you finer control in the composite as well. This will allow you to use them not only as RGB layers to add to the frame, but also as holdout mattes to control desaturation and blur settings for achieving even greater depth. Plus, when all put together, their combined motion will sell the shot far batter than a single, large particle system encompassing the entire scene, whether it is reactive or not.

Bringing it Together

The final scene of the Lexus spot is a long shot looking down an industrial area street. The hero Lexus car drives from under the camera into the distant horizon while three bad guys jetpack after it. The sides of the street are outfitted with set extensions, the day shot is turned into night and everything is ready to be composited together. Layers and layers of particle fog are added to the comp to give the scene depth and continuity. The figure below shows the final frame, layers and all. Note though, its fog layers have been exaggerated here for print purposes. Subtlety is king.

The background fog is layered to intensify further down the perspective of the street. The intermediate and foreground layers are all reactive to the motion of the car and jet packing bad guys. Fields move the particles out of the way as the bad guys swoop into frame and as the car barrels through the fog down the street.

Black holdout objects mask out the sides of the street and make the foreground clearer and more saturated to the eye. Everything is put together in Discreet Inferno to meld the live action off its green screen and into the scene, bad guys chasing the Lexus and their dashed dreams of victory. Go good guy, go!

LEXUS FINAL FRAME.TIF

Gallery

To have your work considered please submit to
www.highend3d.com/artists/add.3d

David Ying

Workplace: Freelance/contractor **E-mail:** yinako_minako@hotmail.com **Highend Username:** yinako
Software used: Maya4(maya4 render)
Special techniques: As you can see, the image (kelivina) has a non-realistic shading applied to the whole scene. The shading was done by building a shading network using standard Maya shaders and texture nodes. The process was speeded up by writing a script that would make such a shading network quickly and more intuitively by tweaking the colors on the objects without the light affecting its colors.

The shading appears more paint-like, because, as you know, human paintings do not have perfect interpolation between adjacent colors; to create this effect, color interpolation is simply broken down into bands of colors (or sometimes slight changes of hue) in those bands.

The character was also built with this type of shading in mind. Each piece of object was colored with this shading, which meant I needed to model each separate part that needed a different color. So there were no textures in the scene, and such details as borders on her clothing were modeled. All scene objects were built with polygons and smoothed before render. The characters was rigged and placed on the sheet, which was modeled by hand and adjusted by a poly sculpt tool.

For a detailed explanation on my shading script refer to: http://www.noboundrees.com/yinako/tut/nprs/nprstut.html

You can also download this script(nprs.mel) at highend.com
http://www.highend3d.com/files/dl.3d?group=melscripts&file_loc=nprs-v0.6-.mel&file_id=1794

Hardware: Atholon 2Ghz **Render time:** about 12 minutes

Michael Sormann

Workplace: Rockstar Vienna

E-mail: sormann@nextra.at

Highend Username: michael sormann

Software used: Maya

Special techniques: Light dome

Hardware: Athlon 1800+ 500Mb ram

Render time: 10 minutes

Michael Wilson

Workplace: Burst TV Ltd : www.burst-tv.net
E-mail: mike@burst-tv.net **Highend Username:** casinowilhelm
Software used: Maya, Photoshop
Special techniques: The 3D was matched to the photograph by using light probe information captured at the shoot, which was converted to a MAYA lighting environment using Paul Debevec's HDshop. "Raydiffuse" was also used for a "dirt" pass that was composited onto the clean raytraced image. Final touching up was done in photoshop. The facial tattoo was real.
Hardware: Athlon 1800xp, 1Gb RAM **Render time:** 16hrs (4000x4000 image for A3)

Alessandro Nardini

Workplace: Frame by Frame Italia s.r.l. - www.hotpost.it
E-mail: alexnardini@tin.it **Highend Username:** alexnardini
Software used: 3ds max - Maya **Special techniques:** Intensive use of CAMERA MAP Modifier
Hardware: Intel and AMD platform **Render time:** It takes about one hour for render.

Li Xiaoyong

Workplace: Screampoint, LLC , Guangzhou Company
E-mail: lxy_3d@etang.com **Highend Username:** killcat
Software used: 3ds max 4.2
Special techniques: render by Finalrender **Hardware:** P4 1.5G 512M SDRAM
Render time: 1.jpg(16Hour) **other:** 1-2Hour

Chuck Chen

Workplace: Taiwan
E-mail: chuck@apfamily.com.tw
Highend Username: Chuck
Software used: Painter 5.5
Image Completion time: 5 hours

Steven Stahlberg

Workplace: Optidigit, Kuala Lumpur, Malaysia
E-mail: stahlber@optidigit.com **Highend Username:** Steven Stahlberg
Software used: Photoshop 5
Special techniques: Hard-edged pressure sensitive brush, Wacom tablet,
Hardware: P3 733 Mhz, 384 Mb RAM
Render time: Not applicable

Pascal Blanché

Workplace:
Ubisoft Montreal

Email:
lobo971@yahoo.com

Highend Username:
pascal blanché

Software used:
3ds max 5.1, VRay renderer, Photoshop.

Special techniques:
Some tricks that were very useful: I used some camera mapping for the details on the body, and all the jewelry was made using some dingbats font splines ;)

Hardware:
PC tower- Pentium III 1.2 GHz - 1 Gig Mo RAM

Render time:
4 hours

Tulay Tetiker

Email:
tt@cine-fx.de

HighendUsername:
poocky

Software used:
Photoshop 7

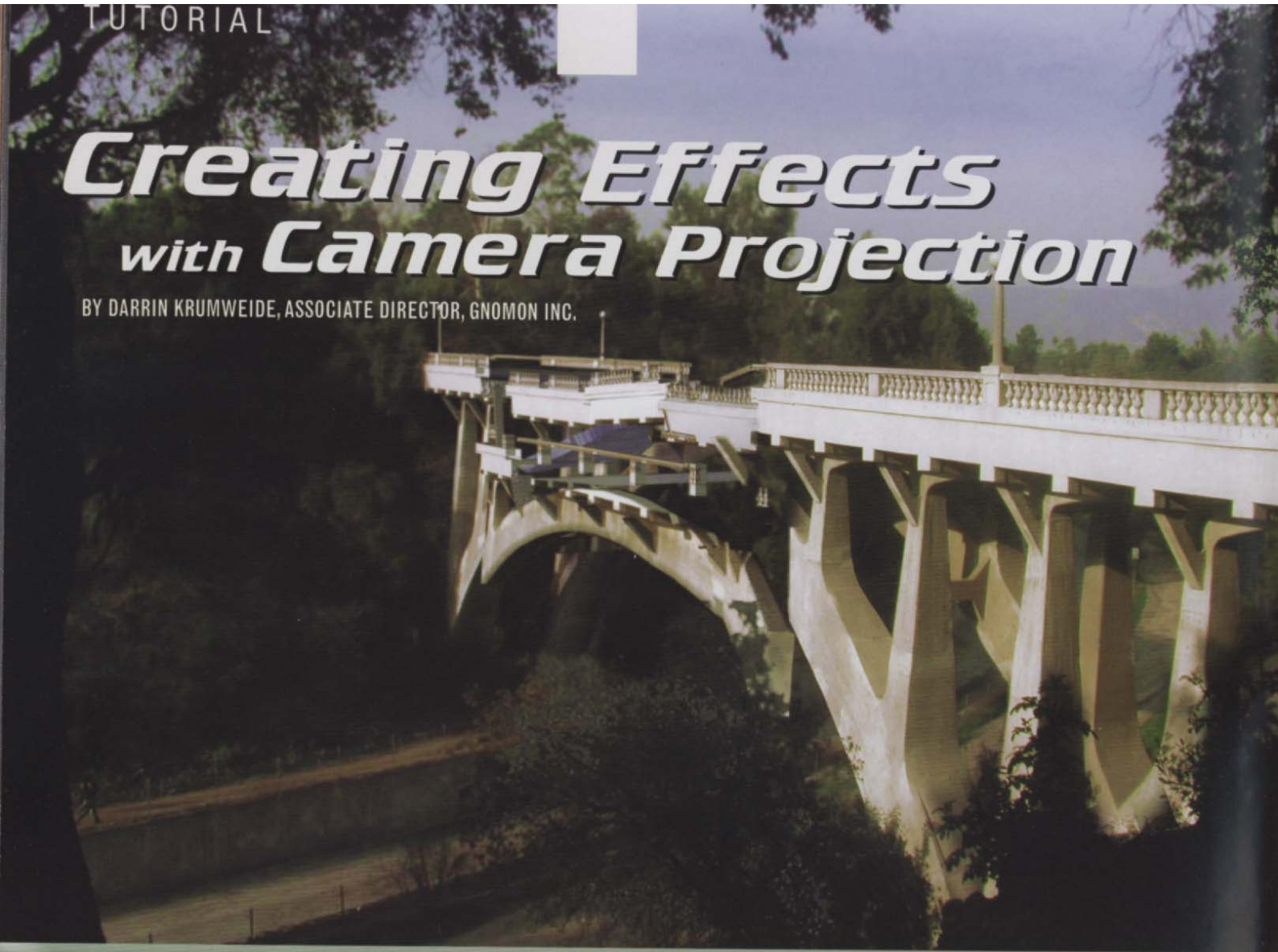
Special Techniques:
Deep Paint, Wacom Tablet

Hardware:
AMD 2,1 Ghz, 512 MB, 80 GB harddisk, Geforce 3 Card, Wacom Intuos Tablet

Image Completion:
1 day

Creating Effects with Camera Projection

BY DARRIN KRUMWEIDE, ASSOCIATE DIRECTOR, GNOMON INC.

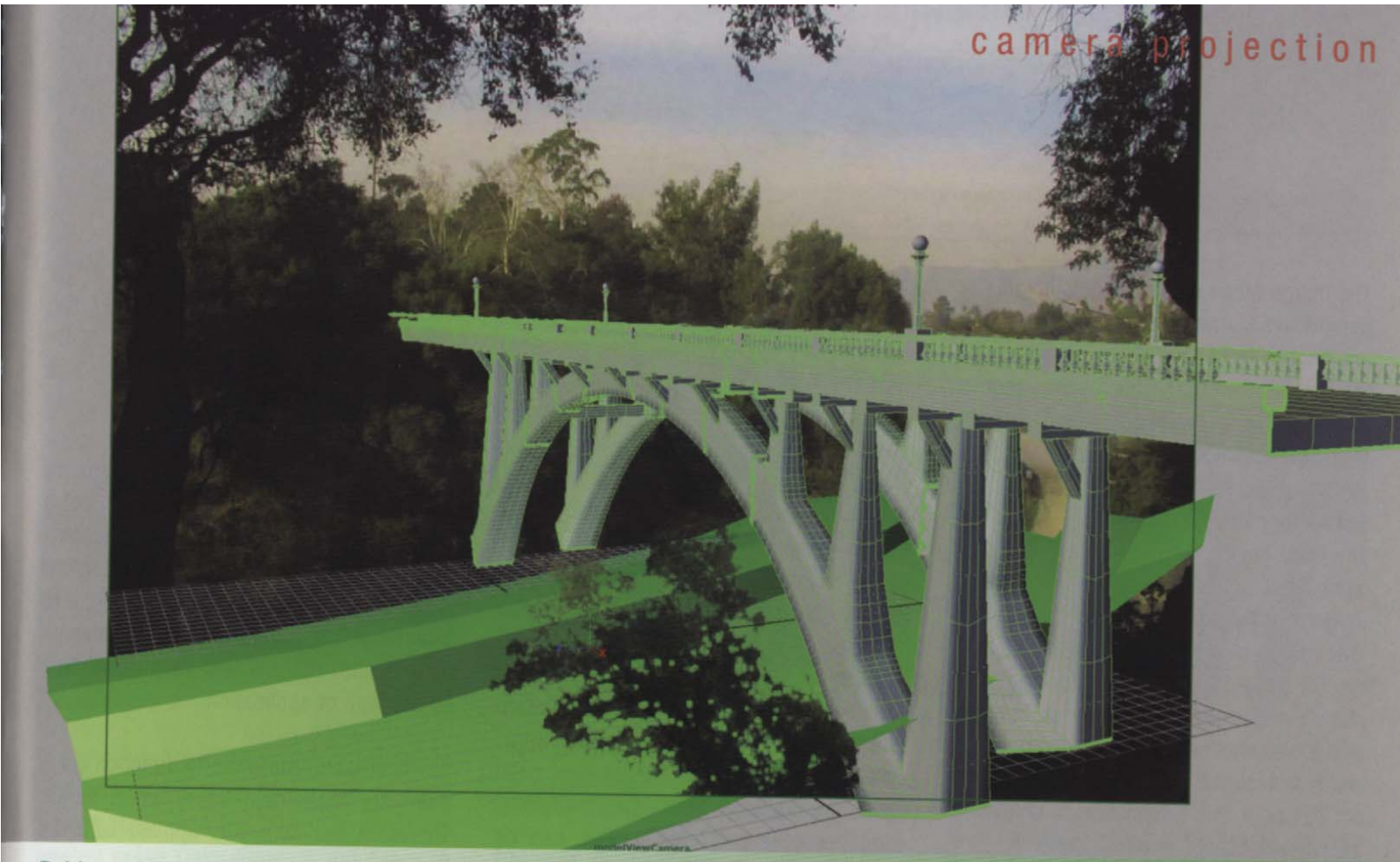


CAMERA PROJECTION is one of several 3D projection types used for placing textures on surfaces. The application of Camera Projection to a shot is a powerful way to add dimension to any matte painting, photograph or live-action background. These techniques are comparably more cost effective and faster than traditional effects. Similar to traditional methods, camera projection allows the artist to create the same miniatures, props, pyrotechnics, and high-speed camera effects without the added expense or consumption of time.

Typically, there are two elements to a Camera Projection scene: background imagery and foreground textured geometry. While most scenes contain both these elements, what makes camera projection unique is that both background and foreground elements use the same image. Because of this, there is no noticeable distinction between them. This blend creates a unified image, yet it also allows the artist to manipulate the shot by changing lighting, animating foreground elements and masking any additional geometry within the scene. For example, the figure above contains a modeled bridge superimposed on an actual bridge. Yet the geometry of the bridge is animated and lit to make it appear as if collapsing.

A background plate of a bridge is altered by modeling a 3D representation of the bridge, camera projecting the background onto the geometry, removing the bridge from the backplate, and animating the 3D bridge collapsing.

When constructing a scene using camera projection, the background image serves as a modeling template for a more accurate placement of foreground geometry. By using the background image as a template, the artist has an absolute reference for surface construction, making the construction itself less demanding than more traditional methods.



Bridge modeled in the foreground matching the background plate

While the background is used as a template as in the above example, it is not locked to the foreground geometry. This is important to note because while the background assists the artist in building and placing a model, the background may be removed any time, its only use then being a template for precise surface modeling. The use of the background, either as an element within the scene or only as a template for building the foreground geometry, is up to the artist. Nevertheless, the artist is assured that the texture projected on the model (in this case, a bridge) will match the surface because it was built from the background template.

A different application of camera projection may help in understanding. If an artist were modeling a character's face from a background template, the background may be removed, leaving only the foreground texture projected onto the face. The added advantage is that the projected image contains the same realistic elements such as color and placement of facial features, while it carries all the same dimensions of the original picture.

In the following discussion, steps for building a camera-projection based effect are covered. Some of the steps use terminology that is general to most 3D packages, while some are specific to tools used in Maya.

What Types of Images Work Well?

Select images depicting architectural elements having minimal details and blocky shapes. Straight lines such as walls, streets and windows make positioning the camera easier. Avoid foreground greenery because it is difficult to mask out and obstructs the view.

Matching Camera and Image Plane.

In the camera set up below, the Camera Gate and image aspect ratio are tied to the Camera's Rendering Resolution as opposed to a Film Gate. Either aspect will work; just make sure that the Render Globals, the image plane fitting, and the Camera Projection are synchronized.

Camera Setup

In the Render Globals–Change the Resolution Height/Width to the source images resolution or a proportionate value.

In the Camera Panel–Enable Resolution Gate in the Camera Settings

Import the back plate image into the perspective camera as an Image Plane

In the Image Plane Attributes - Change the Image Plane Placement,s to ŒFit to Best,

Fit the Image Plane to the entire resolution Gate by selecting Fit to Resolution Gate.

The image plane should completely fill the perspective camera's resolution gate. If the entire gate is not visible on screen, raise the Overscan attribute for the perspective camera. This is located in the Film Back menu of the Camera's Attribute Editor.

Orienting the Camera

The focal length, viewing angle and line of sight must correspond to the background image in order to properly orient the camera. This can be a complicated step of matching the CG camera's perspective to the image plane's perspective. Knowledge of how the image was shot such as focal length, f-stop and lens type will make orienting the camera easier. If this information is unknown, the camera may be approximated.

In the bridge example, blocks are placed in 3D approximately where the location of the front pillars exist. They are then duplicated and moved to the approximate area where the back pillars are located. If the camera's convergence of the pillars does not line up with the image plane, then the focal length is incorrect and should be readjusted.

Once the camera is appropriately placed it should be locked to avoid moving it from its settings. This will be critical once geometry is placed in the scene covering represented parts of the background image.

Modeling

Generally, modeling with polygonal geometry is easier than using NURBS. In the bridge example, the arches are created by lofting two curves together forming the front facade then extruded to create depth.

Modeling surfaces are less difficult when referencing a front or side orthographic view. Nevertheless, the typical view used in camera projection is skewed, thus modeling becomes problematic. This can be fixed by using construction planes oriented to the same perspective view used by the image plane. For example, a construction plane is set in perspective along the side of the bridge. Then curves are drawn by plotting control points along the construction plane, referring to the image plane as a positioning template for each curve.

TIP: MODELING EVERY COMPONENT OF THE SCENE IS UNNECESSARY. ONLY MODEL THE COMPONENTS THAT ARE INFLUENCED BY MASKING, LIGHTING OR ANIMATION.

Texturing

In camera projection, the foreground geometry is textured using the same image as the background. This is accomplished by importing the same texture found in the original shot and applying it to a material. The connection made between texture and material will vary depending on the type of application.

When creating surfaces to function as a geometry mask, lighting is not a factor. Therefore, the texture is applied with full luminance not affected by light. In this case, the texture is connected to the Incandescence Channel.

If the projected texture requires the influence of CG lighting, then the texture is connected to the Color Channel.

Highlighted arch (left) was converted to a file texture after UV coordinates where established (right). Surface textures were then cleared of shadows and shading generated by original image.

Hardware shaded view showing stationary Texture Reference model depicted in gray template

If the geometry is deformed or translated (blowing up a bridge, house, etc. or other deformation) then the texture is also connected to the Color Channel and needs to be frozen or baked onto the surface. This last application was used in the collapsing bridge example.

The following steps create a texture map placed with Camera Projection:

- In the texture editor (Hypershade or Multilister), create a File texture as Projection.
- Connect the projection node to a Lambert Material as Color
- In the Projection Node Attributes, Change the Projection Type to Perspective
- Open the Camera Projection Menu of the Projection Attributes
- Link the projection to the appropriate perspective camera
- Change the Fit Type to the same Fit Type as the Image Plane. In this example the Image Plane is fit to the Resolution Gate, therefore, the Camera Projection is Fit to the Resolution Gate
- Assign material to all surfaces

Polygonal Surfaces

Even though camera projection is a world-based texture placement technique, it is beneficial to establish good UV coordinates for polygonal surfaces. This will give a more accurate depiction of surface texture in Hardware Texturing Display and will be necessary for cleaning up textures later. The fastest way to establish UV coordinates is to apply Automatic Mapping; found in **Edit Polygon > Texture > Automatic Mapping**.

Rendering

The photographic image of the bridge contains surface lighting, shading and shadowing based on the sun's current position in the sky. Any modification to the CG bridge's original position will affect the current representation of all of these. Therefore, artificial lighting is added to the CG bridge and any shadows cast onto the surfaces will need to be recreated by artificial lights. This also means that existing shadows inherited from the projected image are painted out of the texture maps.

New lights are added to the scene mimicking the sunlight's intensity, direction, color and shadow color. Because CG lighting does not account for the light that naturally bounces from any surface onto another, more light needs to be added to mimic this natural occurrence.

Shadows and shading established from the original image are painted out for any key surfaces. Use Convert to File Texture to extract normal textures from the Camera Projected Materials. Each texture can then be brought into Photoshop so that the shadows can be cleared off the map. A 3D Paint program can also be used in addition to or in place of Photoshop.

Remember, before converting projections to file texture, texture coordinates need to be established for all polygonal surfaces.

Convert to File Texture

To create custom UV textures for each polyset or NURBS object, select all surfaces that need cleaning and Shift Select the Projection Node in Hypershade.

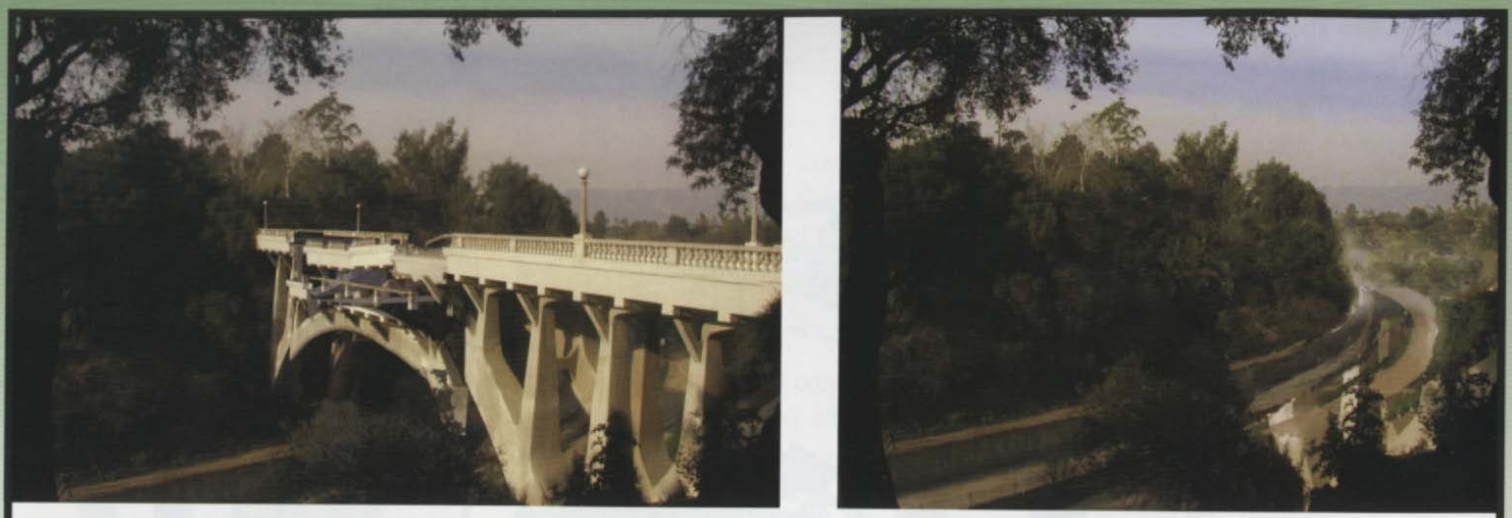


Image on left is the original photo. Image on right is retouched in Photoshop with the bridge removed and background trees and shrubs painted in.

Open the Convert to File Texture options located in the Hypershade's Edit menu.

Reset defaults

Change the file format to accommodate the paint program.

Create Texture Reference

Some of the smaller or more obscure moving objects have less noticeable shadow or lighting inconsistencies. In this case, avoid the creation of several file textures and opt for Create Texture Reference. This command still creates geometry free from texture sliding problems but without the ability to edit textures in paint.

Texture Reference objects are duplicate surfaces that remain stationary and templated while their animated counterparts translate in 3D space free from texture sliding. Always create texture references while objects are in their original positions. All animation is deleted from reference models before rendering.

Animation

Surfaces with camera-projected textures may be used as active participants of an animation. This gives the appearance that components of the background image come alive. The back plate itself is inanimate; however, the surfaces representing components of the image can translate and deform when animated to do so.

If camera-projected surfaces are animated, then the integrity of texture placement becomes a factor. An artist will want to avoid a texture sliding, an effect caused by spatial changes between geometry and texture placement. Additionally, the back plate or background image needs consideration. If a camera-projected surface moves through space and presumably carries its texture with it, a void is created where the animated object was originally placed. This will require replacing or filling the void in the background with the appropriate environment. In the above figures, Photoshop is used to remove the bridge from the back plate.

This tutorial describes only a few of the many ways Camera Projection may be used. As with all things, practice is the best way to gain an applicable understanding. To better appreciate Camera Projections and their application, it is best to try incorporating these methods on your own shot.

If you are looking for a forum to discuss and experiment with these issues, Gnomon hosts an ongoing series of Visual Effects challenges at Cgtalk.com. The VFX challenge focuses on supplying participants with the rules and guidelines necessary to execute an effect which integrates with a live-action background plate. Camera Projections have been a key requirement to dealing with past challenges.

Demonslayer

BY KAT CURRY, CRAIG SYLVESTER, KIRK SHINTANI AND PAM HOGARTH

Our modeling team produced many different spiders, three of which were chosen for the shot.

IN EARLY 2002, GNOMON was approached by the Los Angeles Film School to work on a collaborative project, a Roger Corman film that would require several visual effects shots. After some exploratory meetings and permission from software vendors (Maya, Shake, Rayz) to use Gnomon's educational licenses, we signed on.

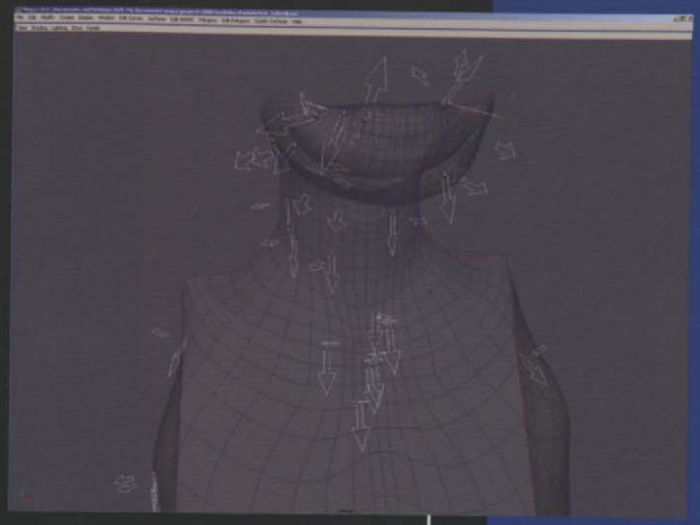
The film, "Demonslayer," had your typical B-movie plot: Contemporary juvenile delinquents are sentenced to clean up an old house for their community service. Of course, the house was built upon the site of something old and evil; in this case, a Wild West whorehouse. While completing their community service, the juveniles become possessed by the spirits of the dead prostitutes and all hell breaks loose…until the local priest saves them.

Although "Demonslayer" was a student project, we wanted it to be as much like a real-world project as possible. We enlisted the help of veteran digital artist Gil Zimmerman (currently Head of Layout, DreamWorks Feature Animation), as Visual Effects Supervisor. CG Supervisor Craig Sylvester handled day-to-day supervision. Twenty-some Gnomon students then signed on for the project and were organized into departments such as modeling, texturing, and so on.

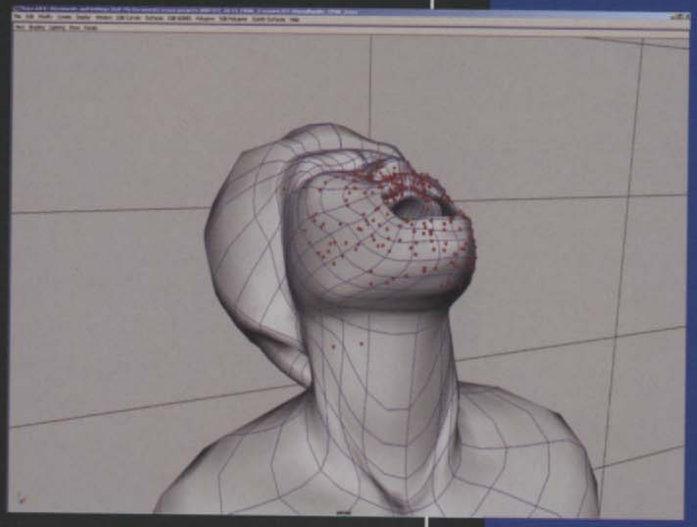
Sylvester met extensively with the film's Director and Production Coordinator to determine that there were approximately 60 shots to be done. These would involve the creation of CG spiders, cockroaches and snakes, a digital set of the whorehouse exterior, flashback "vision" sequences and spirit "presence" sequences. A majority of these were 2D effects and simple composites, but three scenes called for extensive 3D work. The film was shot by the LA Film School production company in HD with a Panasonic 24p camera. Cinesite Digital generously provided the tape-to-digital-and-back-to-tape conversions.

Of all of the shots we signed on to do, the Tamara spider shot was the most complex. Everyone involved in this shot was pushing new ground in their work, not only in the technical concerns of their department, but in integrating their work with the needs and contributions of the other departments.

The shot was described very loosely to us as "a demonic woman runs down a dark hallway, stops in front of the camera and a large spider bursts from her mouth." Initially, lacking footage or even storyboards, the CG artists had complete creative license. After some discussion, it was decided that a host of smaller spiders would stream out of her mouth as well. The grand entrance of the "queen" spider would follow the swarm of spider minions.

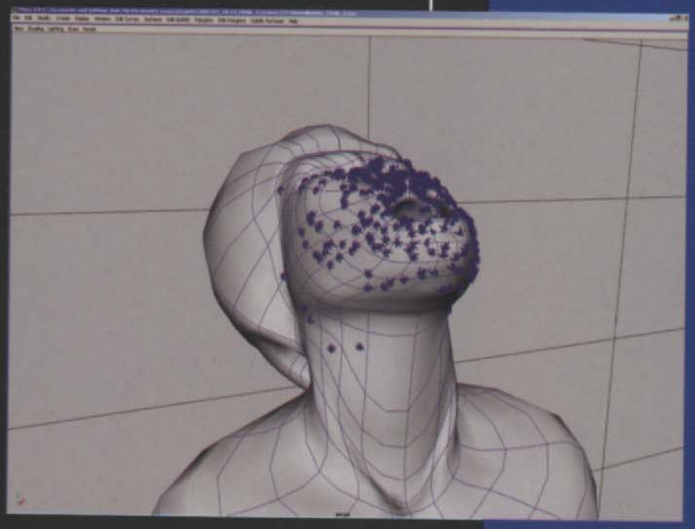


Instanced arrows for vector reference



Designing surface particle flow

Instanced spiders



After this idea was approved by Zimmerman, the different teams went to work on their aspects of the shot so that we'd all be as prepared as possible when the actual frames arrived. Motion Team Lead Kat Curry gathered a large number of spider photo and video references for animation and character setup. While our character animator Wade Middleton spent time studying the locomotion and personality of many types of spiders, Curry was able to experiment with different motion controls.

Meanwhile, the modeling department, headed by Duane Rutkowski, began sculpting a NURBS patch proxy-model of the actor's head and shoulders. This proxy-model would be track-animated to the footage and provide a surface for our spiders to crawl down. It would also occlude the render of spiders in her mouth. The modeling team's only reference of the actor was the footage we were receiving for FX and, as happens in horror films, much of the footage was of poorly lit hallways. The model and tracking had to be very accurate to the actor, so that the spiders wouldn't "skate" on her skin or fail to register with the shifting contours of her face. Not only did the lack of data on the actor, set and camera make the proxy-model difficult, it also made the use of tracking software impossible. Sylvester was forced to manually roto-animate the proxy-model from the actor in the shot.

The particles/dynamics team determined that a single NURBS-surface proxy-model of the actor would provide the best UV space for animating particle-instanced spiders. Point man, Kirk Shintani, leveraged advice from experienced mentors, including Zoic Studios co-founder and Gnomon instructor Andrew Orloff.

By default, Maya flips a particle's up-vector on a surface when goaled if the particle's up-vector orients itself at too great of an angle from world up-vector. Orloff explained the pointOnSurfaceInfo node and was instrumental in getting this effect to work. By using a pointOnSurfaceInfo node, we were able to force the particles to maintain a constant up vector. Creating a pointOnSurfaceInfo node required us to type "createNode pointOnSurfaceInfo" into the command line. Once created, the node was attached to the surface by connecting the worldspace of the shape node of the object to the input surface of the pointOnSurfaceInfo node. In order to see the pointOnSurfaceInfo node in the outliner, we turned off the "DAG objects only" filter under "display." To maintain a constant up-vector, in runtime, variables were created to hold the goalU and V values. These values were then passed to the pointOnSurfaceInfo node by using the setAttr command for the nodes' parameterU and V values. By assigning these values to the pointOnSurfaceInfo node, it will return an XYZ vector for the normal at every frame of the particle's movement on the surface. We then cre-

Spider Beauty Pass

Snakes

ated three float values to hold the X, Y and Z values, used the "getAttr" command for the normalizedNormal of the pointOnSurfaceInfo node for X, Y and Z, and then used these floats in a vector for a custom attribute called "myNormal." This attribute was called in the particle instancer section of the particle shape to orient the AimWorldUp.

Now that the surface was prepared, we had to create expressions to control movement along the surface. These expressions began with simple goalU creation expressions that established random rates for the particles along the surface. These were then called in the runtime so the particles would actually move down the U direction of the surface. The value for goalV was also randomized to add movement in the V direction so they would not appear to just follow the surface parameterization. We also created expressions to kill off particles when they were no longer visible to the shot camera to reduce the number of instanced spiders on screen at any one time, and to prevent sticking along the edges of the surface.

For the instancer, the spider model was animated in a cycle. Spider geometry was duplicated in every frame of the

Vision Sequence

animation, (there are scripts that will do this automatically such as "instCopy" by David Parisi, available at Highend3D.com) selected in order and then instanced to the particles. (Note: This method of instancing prevents the use of motion blur on the instanced geometry because the geometry is only visible for one frame at a time as it cycles through all the frames of the animation.) Variables and expressions were created to use age, velocity and magnitude to control the speed at which the animation played back based on the speed of the particle. A random start expression was created to prevent the instanced spiders from cycling on the same frames. Also, keep in mind that the instanced geometry should be VERY light. No patch or high-resolution poly models here. The spiders we used were very low-res because their size in the overall frame didn't require anything more detailed, and because of the number of particles on screen at one time.

Our modeling team produced many different spiders, three of which were chosen for the shot. Jericho Green provided a low-poly, jumping spider that had a strong silhouette. This was used for instancing onto the actor's face. The tarantula modeled by Bret St Amour was scaled down to a "medium" spider, and motion-pathed into the swarm of small spiders to provide variation. Tony Carrillo modeled the "hero" spider for the shot. This epic King Crab/Black Widow-like spider slowly squeezed out of the actor's mouth as she threw back her head.

Once the spider models were finalled, Curry adapted her rigs to the different spider types and Middleton began the animation. Although each type of spider would move differently due to the disparate leg proportions, the real variations would be animation choices such as timing and weight. Despite the differences of proportion in the spiders, Curry was able to adapt a single complex rig to each. All the spider controls and surfaces were grouped under one "super node," so that the "local space" of that node could be moved or scaled to fit in the scene without affecting the animation. This allowed us to parent the spider to the proxy's jaw joint, so that the spider's

movement would be relative to the animated proxy.

With final spider models in hand, Bonnie Rosenstein and Forbes Hill, the two texture artists, began the lengthy process of development, feedback and approval for all the spider textures. Every UV and isoparm tweak required that other teams update any models they'd been working with.

Finally, all the teams had approval on their separate aspects. It took several adjustments from each department for all the elements to be gathered into single scene files. With models, rough animation and textures together, lighting tests revealed that the hero spider textures didn't read well in the darkness of the shot.

Fixes bounced back and forth between lighters and texture artists to finalize the look of the spider. Meanwhile, adjustments were being made to the hero spider animation to get the right feel. Now that the particle spiders could be slap-comped against the back plate, their flow and timing were also being tweaked.

With the 3D elements completed, our attention turned to the 2D concerns of rendering all the passes and compositing the shot together. We quickly discovered that, with the complexity of this particular shot and the overall demands of rendering thousands of 1920x1080 frames, we were under-prepared for the demands of the end run.

We needed storage and more rendering capability and we needed them now. Our FX Producer Pam Hogarth was able to get donations of the needed resources. RFX provided consulting services and 300 Gigs of external storage. EPowerGate extended an offer to render footage in exchange for feedback on how their Renman system handled HD Maya renders. Access to their external render farm helped speed render times and relieved the demands on Gnomon's computers.

Former Gnomon student, Wensen Ho, and veteran artist, Lawrence Huang, volunteered to take on much of the project compositing. Their extensive Shake experience made a big difference in the outcome of this, our most complex, shot. The final composite consisted of beauty, specular and shadow passes for each type of spider and some extra variations for the hero spider.

The coordination and cooperation it took to successfully complete this shot tested the flexibility and resolve of each one of us. Of course, when you're sitting in the dark with the other people who shared your pain and you see your work projected on the big screen, it's all worth it.

Presence

"KUNG POW - ENTER THE FIST"

A compositing confessional

BY JOHN LAFAUCE, JR.

THE STAGE IS SET

After a family is suddenly murdered while enjoying an intimate dinner in their home, Master "Betty" Pain and his evil henchmen realize their work is not yet finished. They approach a crib and inside it, an infant known only as "The Chosen One," is cooing innocently. Master Pain approaches the crib and marvels at his discovery. His smile fades as he slowly draws a huge knife from the sheath at his side and raises it above the infant. In a flash, the knife lunges down, only to be abruptly stopped by the infant's tiny hand. The baby is unusually strong, with a firm grasp on Master Pain's wrist, which cannot break free. Suddenly, little booties appear and in a blur, Master Pain is kicked numerous times in the face. The infant springs from his crib and proceeds to wreak martial arts havoc, displaying amazing acrobatics and fighting prowess. Flipping through the air, running up walls, rolling on the floor, could this mere baby be a DNA clone of Van Damme himself, with possibly the Three Stooges thrown in to give him personality?

Master Pain is not deterred by this show and he proceeds to lash wildly at the infant, who deftly and playfully ducks his every move. Master Pain's confidence turns to frustration as a near-miss swipe of his sword causes the Chosen One's diaper to be cut off and fall to his feet. In typical baby fashion, a urine spray showers Master Pain and all the Chosen One can do is giggle. The baby escapes, Master Pain burns down the home and so begins this 80-minute quest by The Chosen One to avenge the death of his family.

Welcome to "Kung Pow–Enter the Fist."

As you can see from my introduction, this wacky story is a far cry from high art and is especially not your typical effects-driven film. But, it is a spoof with many contemporary references which promises to make you laugh while it displays a wide range of effects trickery that is impressive artistically, technically and in its scope.

fig 1 Original footage from "Tiger and Crane Fist," 1976

fig 2 Bluescreen figure of "The Chosen One" (Steve Oedekerk)

fig 3 Head is replaced for "Kung Pow - Enter the Fist" 2002

HUGE AMOUNT OF EFFECTS

"Kung Pow" is the brainchild of Steve Oedekerk, who serves as the writer, director, producer and star of the film (He plays The Chosen One). The film was released by Twentieth Century Fox last year and has since garnered shining reviews for its effects, especially for the computer-generated baby fight sequence at the beginning of the film and the "Matrix"-like Kung Fu showdown Oedekerk has with a CG cow.

Chris Watts, Visual Effects Supervisor and Betsy Paterson, Digital Effects Supervisor of Dubbed Films Visual Effects, helmed the bulk of effects work for this film, which features over 2,000 effects shots. To their credit, this film was an enormous undertaking and showcases not only some fine character animation and CGI compositing, but also a large amount of "invisible" live action compositing as well. It is this meticulous work that a talented and tireless artistic team of compositors, painters and rotoscopers worked for many months to complete. I was a lead compositor on that team and I'd like to give you a demo on one of these shots.

fig 4 Clean plate pre-comp with articulate collar roto

fig 5 Primatte with soft, natural matte edge

fig 6 Primatte as an inside hold out matte (HOM)

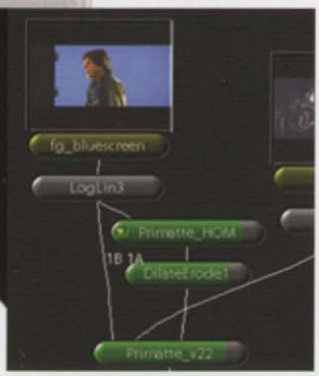


fig 7 Tree detail of Primatte arrangement

HEAD AND BODY REPLACEMENT

Our task was to substitute an actor from a 1976 Hong Kong action film, "Tiger and Crane Fist," (see fig. 1) with a present day actor (Oedekerk) shot against bluescreen (fig. 2). In modern moviemaking, faces and heads of actors are routinely composited over their stunt doubles and even CG stunt doubles are used where it would be too dangerous or impossible to film a scene using human actors. These are not new concepts. What makes "Kung Pow" so unique is that a serious actor is being entirely replaced by a comedic actor who assumes the lead role of the film (fig. 3).

The scene I'd like to demo is a head replacement comp I did when the Chosen One and a female character named Ling (Ling Ling Tse) are bonding and eventually embrace each other. To meet the length requirements of this magazine article, I cannot accurately explain every step in the compositing of this shot. So, I will gloss over much of the routine setting up of the comp and focus on keying and integration tricks I used to make this shot even more believable. The areas that I will illustrate in more detail will be:

- Combining two Photron Primatte nodes for natural-looking edges
- Color correction and concatenation
- Blending hair blacks into the bg with a grad

(ALL METHODS AND TOOLS MENTIONED PERTAIN TO SHAKE, UNLESS OTHERWISE SPECIFIED)

BUILDING THE COMP

The elements that make up this composite originated as Kodak Cineon files, which are in 10-bit log color space. The first step of this composite was to convert these images to linear color space using the LogLin node.

Step 1)

Removing the head of the original actor— It was necessary to remove the head of the original actor before comping the new head. A painted "clean plate" frame was provided to me by one of our painters using Puffin's Commotion software. This clean frame was matchmoved to the wall of the bg and grain was added to match existing grain. Articulate roto using RotoShape along the man's collar (fig. 4) and loosely around his head gave us the garbage matte. Additional articulate roto around Tse's head and face were needed to provide a holdout matte. I ran this out as a pre-comp and brought it back into Shake.

Step 2)

Stabilizing the bluescreen fg—Since Oedekerk's performance was very frenetic and animated, stabilizing his head was tricky at times. Watts did a fine job in attaching tracking markers to him for this purpose, as well as using video pre-vis of scenes from the original film to block out his position and movement.

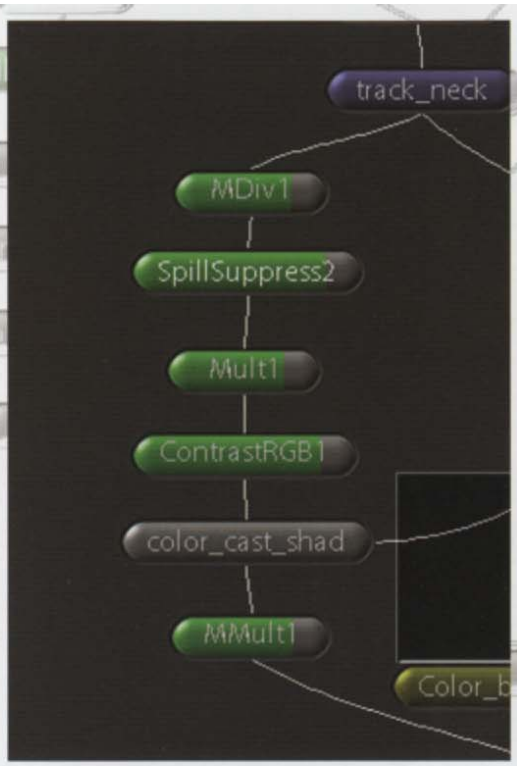


fig 8 Steps to color correct this premultiplied image

fig 11 Grad matte to blend head blacks into the background blacks

fig 9/10 Shadow casting on face of original. New cast shadow added to match

Step 3)

Combining two Photron Primatte nodes for natural looking edges—For this bluescreen, I got great edges by combining two Primatte pulls. My goal of the first pull was to maintain the softness of his edges, especially his hair. Ensuring that there were subtle grays on his matte edges made it look more natural (fig. 5). I achieved this by sampling from the bluescreen backing and scrubbing with the background operator in Primatte to make the blacks of the matte black. I did not use the fg operator to scrub over Oedekerk's head key to clean it up and make it all white. Doing so would fatten his matte, resulting in a harder edge and matte line.

To clean up and correct for the holes in his matte and make it perfectly white, I copied and pasted the Primatte I had just made, scrubbed the contaminated fg matte areas and made it a hard-edged black and white matte (fig. 6). This was used as a HOM (holdout matte) and plugged into the original Primatte's holdout matte input. A DilateErode (fig. 7) was used to shrink this matte slightly, allowing the nice gray edges to appear from the first Primatte pull. A blur can be added to soften the holdout matte edge if necessary. The Primatte output is set to onBlack to premultiply the fg against black.

Step 4)

Matchmoving the head to the body—I could now attach the stabilized, premultiplied fg head to the headless body of the bg. I typically like to view my progress interactively, so I'll temporarily plug the fg and bg into an Over node while I work. For the body track, I found a reference point at the base of the original man's neck. I then applied a Move2d after the stabilization to repo and scale Oedekerk's head to fit into place. It was then matchmoved to the man's body movement by using a Stabilize node with the inverseTransform parameter set to "match." The Move2d and Stabilize nodes 'concatenate' together (I'll mention the benefits of concatenation when I discuss color correction.)

Step 5)

Color correction—This is one of the most important aspects of compositing and color correction in Shake; it is a science in itself. Aside from having a good eye for color matching, a thorough understanding of premultiplication and how it affects color and edges is another essential skill. Since the fg head became premultiplied by setting Primatte's output to On Black, it needed to be unpremultiplied with an MDiv node so color adjustments could be made with optimal results. SpillSuppress was then applied to remove blue spill and Mult (for color) and ContrastRGB (for blacks) were used to color correct Oedekerk's head to fit with the bg. Mult and ContrastRGB are

(IT IS ALSO WORTH NOTING THAT BETSY PATERSON TOOK ADVANTAGE OF SHAKE'S CUSTOM MACRO CAPABILITIES AND DEVELOPED ONE TO MAKE THE NEW FOOTAGE COLOR APPEAR DEGRADED TO MATCH THE OLD FOOTAGE. THIS MACRO WAS DISTRIBUTED TO THE COMPOSITING TEAM SO ALL THE SHOTS WOULD LOOK CONSISTENT.)

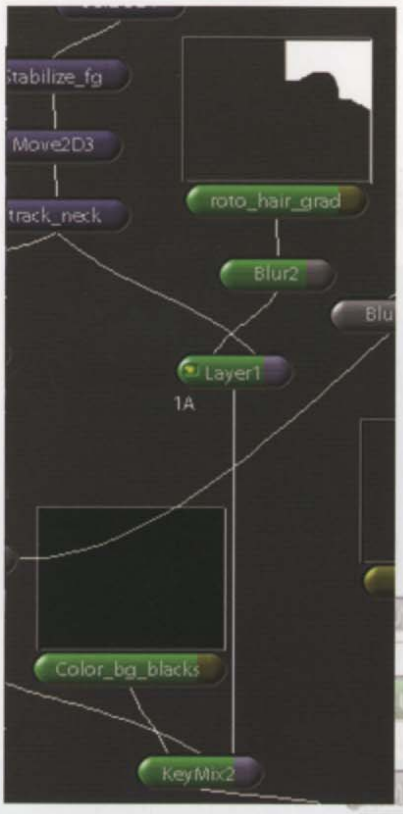


fig 12 Detail of hair grad matte and comp setup

fig 13 In a touching moment, Ling proclaims her love for *The Chosen One* by saying "Take me man-meat."

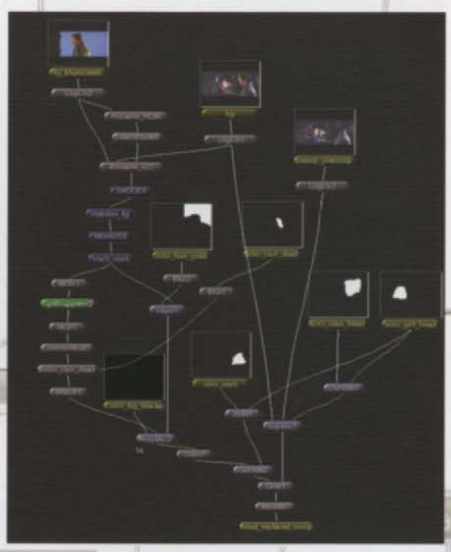


fig 14 Shake's node tree layout of finished head replacement composite

"concatenating" operators that collapse together into one function, allowing for more efficient processing and maintaining of color integrity. An MMult node is applied after the color operators to premultiply the fg again (fig. 8). This can also be achieved later on when it is ready to be composited over the background and the premultiply parameter in the Over node is turned on.

Step 6)

Adding a cast shadow onto the face—The original plate had Tse casting a shadow onto the man's face (fig. 9). I wanted to do the same thing to Oedekerk's face to give the impression that he really is in that environment. I used a simple Rotoshape that was blurred and plugged into a Brightness node to darken that area (see fig. 10).

Step 7)

Blending hair blacks into the bg with a grad—The original plate also revealed to me that both Tse's and the man's hair edges and blacks seemed to "melt" into the bg with no apparent seam. So, I created a roto along the inside of Oedekerk's hair edge, blurred it and performed a boolean with a Layer node set to Inside to establish the grad matte (fig. 11). I took out a Color input node and sampled the dark bg color behind Oedekerk and pumped that through a KeyMix along with the grad matte I just created. I dialed in the fg percent just slightly and Oedekerk's hair blacks and edge blended into the bg nicely (fig. 12).

Step 8)

Matching grain and comping—It's a good idea to add grain after all of the fg manipulations are done. Now the fg was ready to be plugged into an Over node so it could be comped with the removed head bg. Various Rotoshapes were done earlier in this composite to remove unwanted garbage and allow Tse's head to go in front of Oedekerk's as they embrace. (fig. 13)

Step 9)

Final steps before render—If the shot is being recorded to film, a LogLin color node set to 'lin to log' needs to be applied so the comp is converted back to Cineon log color space. If it is for video preview as my demo is, the present linear color space is fine and no conversion in necessary.

It is a pet peeve I have, but I like to strip the alpha off and only render the RGB image. I achieve this by using a Reorder node and setting it to RGBN. A FileOut node is attached lastly and the shot can be rendered. The comp is finished. (fig. 14)

Artist Spotlight
diego avesani

Name: Diego Avesani
Age: 27
Family Status: Single
City of Residence: Brazil / Brasìlia
Length of Time Doing Visual Effects: since 1996
Past Workplaces: I purchased an indy (SGI) with poweranimator (Alias|Wavefront) in 1996 and started studying in my room. Later I opened my own business, Illusion Graphics LTDA, which does mainly advertising projects.

<continued>

1) What are your favorite movies?

My favorite movies are "Star Wars," "ALIENS," "AKIRA," "The Matrix," "Dune," "Band of Brothers," etc.

2) What is your focus/position at your current job?

It depends on the project I'm working on. I coordinate the projects, but always work on the lighting and composition. The truth is I enjoy participating in the entire process.

3) How do you conceptualize your work/ideas?

At first, I try to analyze which is the best way to start; for example, will we use Nurbs, polygons or subDivision. Then comes the texture, animation and always towards the end, I do what I enjoy the most, which is the lighting and shaders. The ideas are always on my mind — even when I'm driving my car or jumping in the pool, they always show up.

4) How do you cope with challenges and roadblocks when trying to create? realistic images?

Observe the real environment, after all, our head is the best realistic creator that exists in real time :). When I have a problem, I always go to forums, talk to photographers about light and so on. The big problem is when I look for some resource that the software doesn't have, but I always find a decent solution.

5) Do you feel satisfied with your results when attempting to create? realistic images?

No, I always think the result is bad. I started studying 3D in my room and when there were no more books to read about the subject, and the Internet was way too basic and in the city I live in there are almost no 3D producers. I knew that the people in So. Paulo, like Alceu Baptiso, were years ahead of me, so my learning process with the 3D was a bit slow because of the lack of information and difficult access to the techniques. I always study alone, but seek out people that work with 3D to trade ideas and techniques. Today I enjoy Live Action, in (rocket) image. I think that was my best result so far. I was able to do it all inside Maya, without using other ENGINE RENDER or composition software. Now I'm studying mental ray for Maya and I think that I'll be able to achieve better results.

6) What are your hobbies or favorite things to do when you're not working?

Doing lab with the Maya and listening to music, every Saturday afternoon, I play Paintball, and go to the movies and play computer games.

7) What was your dream when you started in 3D?

Be able to build my own world and transform it all into a sitcom; tell stories and dreams.

8) What is your take on the current status of the 3D animation industry?

It's growing a lot, especially in the movies. Today, the screenwriters and director can have more creative liberties but I think that there is still lots of limitations, especially the time it takes is the worst thing to me, because it doesn't depend on me anymore, it depends on the machine.

9) What advice would you give to all the college students, aspiring programmers and artists coming into the industry?

Study a lot and take advantage of the people that you know who know the software so that you can trade information. The software has to be a mere pencil to an artist; the most important part is the personal imagination. One can do everything, but many others do it a lot better, so always look for a company that believes in the projects and not the routine.

10) Do you have any philosophies that you live by, helping make you the person you are?

I always try to do my best as a citizen of Earth.

11) What would you like to see added to 3D software?

I use the Maya, I wouldn't trade it for any other software, but would love to see some improvements in the SubDivision, and especially in the time problem, have the same resources as the exterior engine like mental ray, brazil or renderman.

12) What is your take on 3D actors in film?

It's one more area that is evolving, but I think the biggest problem is the animation, the characters always look like they are in a different "timer" as the real actors.

13) Where do you find inspiration?

I watch a lot of movies, designer sites, HR GIGER books, in my own childhood. I was greatly influenced by "Star Wars," but I really like listening to music — the inspiration comes fast!

14) Do you want to describe your short movie that you are working on?

I have two projects in a study phase. One is in cartoon style and is funnier and all and I want to do it all in live-action. It's going to be a small character in relation to earth and a very curious character. I will start filming images next month. And the other is about a sentinel that lives inside the veins of a pregnant woman and will protect the fetus against a virus that will be produced by the mother. I only designed the sentinel, but am still trying to create an environment more underground. The veins are like sewage tunnels, but the script is still being finalized. These are personal projects that I intend to do with my own techniques.

SUBSCRIBE NOW!

HIGHEN
MAGAZIN

HIGHEND MAGAZINE — STANDARD

- ❍ **Highend USA 1 year (4 issues)** $45.00
- ❍ **Highend Canada/Mexico 1 year** $50.00
- ❍ **Highend All Other Countries 1 year** $50.00

HIGHEND MAGAZINE — RUSH

- ❍ **Highend USA 1 year (4 issues)** $60.00
- ❍ **Highend Canada/Mexico 1 year** $65.00
- ❍ **Highend All Other Countries 1 year** $50.00

KEYFRAME MAGAZINE — STANDARD

- ❍ **Keyframe USA 1 year** (6 issues) 54.00
- ❍ **Keyframe Canada/Mexico 1 year** $72.00
- ❍ **Keyframe All Other Countries 1 year** . . $84.00

Keyframe Magazine, created for the LIghtWave community, offers a solid collection of professional-level tutorials, unbiased an in-depth product reviews, high-profile, industry case studies, and feature articles of lasting value.

SUBSCRIPTION & RENEWAL FORM

Name _____

Company _____

Delivery Address* _____

_____

Phone _____ **Fax** _____

Email _____

*If the Delivery Address is in Nevada, please include the 7.25% Nevada sales tax.

❍ **Charge my card** ❑ Visa ❑ MasterCard ❑ American Express ❑ DiscoverCard

Card # | | | | | | | | | | | | | | | | |

Expiration Date ____ / ____

Cardholder Signature _____

Name on credit card (if different) _____

Credit card billing address (if different) _____

_____

Cardholder email address (if different) _____

❍ I'm enclosing a check or money order in U.S. Dollars. (payable to **DMG Publishing**)

ONLINE:
www.dmgpublishing.com

FAX:
Fax 1-702-616-9647

PHONE:
Tel 1-702-990-8656 or 1-888-77

MAIL:
DMG Publishing
2756 N Green Valley Pkwy # 2
Henderson, NV 89014-2120 USA

EMAIL:
info@dmgpublishing.com

GNOMON

WORKSHOP

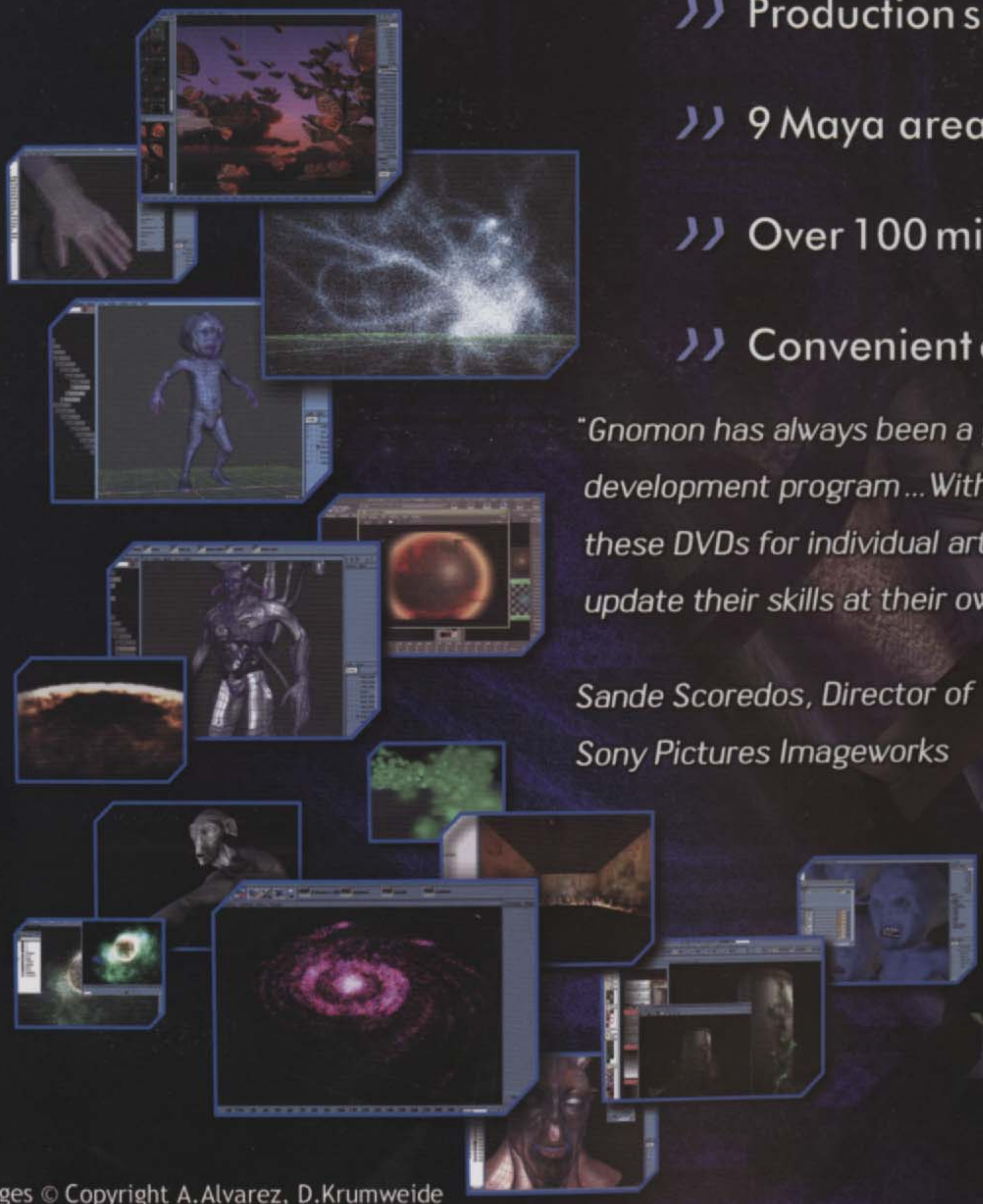
Training DVDs, Tutorials and Tips

Modeling · Texturing · Lighting · Rendering · Animation

Kinematics · Constraints · Dynamics · Skinning

The Gnomon Workshop exists to provide the Maya community with training DVDs and written tutorials. There are currently 38 DVD titles available for ordering.

- Based on Gnomon School curriculum
- Created by industry professionals
- Production specific tips and techniques
- 9 Maya areas of focus
- Over 100 minutes of instruction each
- Convenient online ordering



"Gnomon has always been a great resource for our in-house artist development program ... With our busy schedules, we often use these DVDs for individual artists to explore new ideas and quickly update their skills at their own pace"

Sande Scoredos, Director of Training
Sony Pictures Imageworks

Maya

www.thegnomonworkshop.com