Gama Network Presents:

Gamasutra.com

Event Wrap-Up: SIGGRAPH 2004

By Morgan McGuire
Gamasutra
August 30, 2004

URL: http://www.gamasutra.com/features/20040830/mcguire_01.shtml

Over 27,000 game developers, film industry professionals, and scientists attended SIGGRAPH 2004 at the Los Angeles Convention Center from August 8th to the 12th, 2004, to discuss the latest developments in 3D graphics. This SIGGRAPH's forecast for the future of game development: radically more programmable graphics cards, smart and flexible art tools, and real 3D displays.

Several events ran simultaneously during the week at the conference, forming tracks for all aspects of game development. For example, visual art insiders reviewed new films and cut-scenes. Even those who didn't understand the details could ogle the amazing effects. This year's SIGGRAPH favorites include the *Onimusha 3* cut-scene with the ninja boss, shown theater size, and Pixar's new Boundin' short with a Jackalope hero.

Artists also checked out the Guerrilla Studio to experiment with the latest tools for free, including motion capture and beefy workstations loaded up with MAX and other modelers. For particularly abstract creative types, there was even a CyberFashion Show which displayed cyberpunk-esque fashions reminiscent of Giger and The Matrix.

The Psymbiote, Host of the 3rd Annual SIGGRAPH CyberFashion Show - photo courtesy of psymbiote.org

SIGGRAPH and the Importance Of Graphical Research

With games like *Far Cry*, *Doom 3*, *Half-Life 2*, and *S.T.A.L.K.E.R.* hitting shelves this year, one might wonder what is left to research in the graphical arena. Why should we track seemingly theoretical graphics research in the games industry?

Well, obviously, despite the visual success of today's games, there's a lot of work left in computer graphics. Although real-time rendering has made incredible strides, look at the content creation process and the run-time animation in your current product. If your studio is like most, creation aspects are mostly unchanged in the last 5-10 years. Artists tweak vertices around to create models, a time-consuming way of translating their vision into data. Those models then appear in the game mostly as pre-animated set pieces. But even with great AI, characters look stupid when their dead eyes gaze into the distance or they fail to shift their weight walking over rubble.

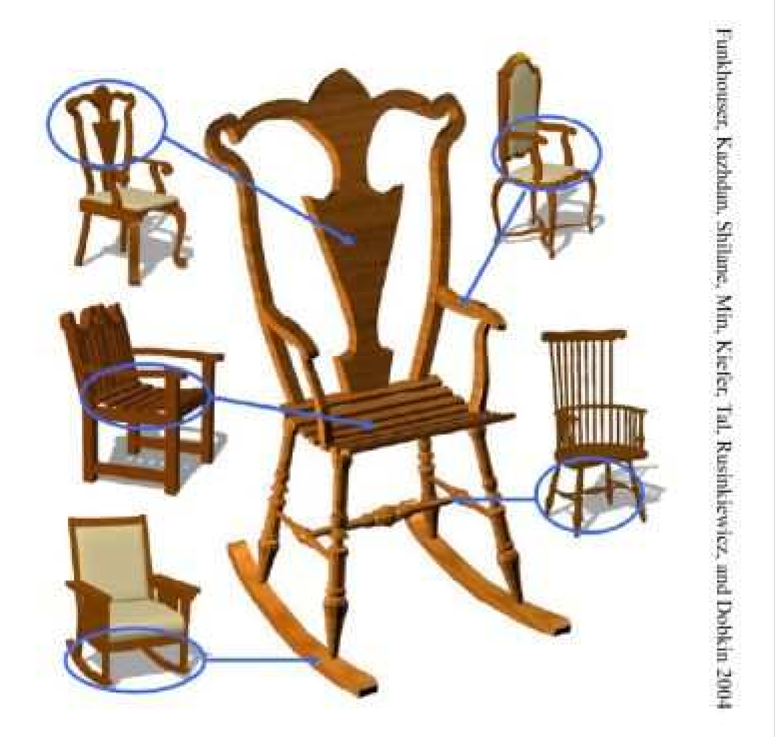
So, with the lion's share of budget, schedule, and head count for new games devoted to content creation, many are trying out ways to make the art process easier, cheaper, and faster. We should also get the most out of every asset by letting gamers interact with them in new ways. Let's look at some of the new ideas from the research community that will make this possible.

Mesh Editing

Artists create models by deforming basic shapes, and by piecing together previous models and carefully fixing seams. It takes years to learn to make the first model and days to create the hundredth. The process is time-consuming because it forces the artist to work with vertices and texture coordinates.

But imagine an editor in which you instead start creating a rocking chair model by searching a database for "chair" and a 3-stroke sketch that looks like a lower-case "h". The search engine returns hundreds of chairs -- you select the back of one, the legs of another, the seat from a third, and so on. The editor then intelligently cuts the various parts from the original models, normalizes the scale and merges them into a seamless new mesh.

This is the powerful new vision offered by the Modeling by Example project. The current system can do all of the above and is usable by both professional artists and amateurs (now "programmer art" doesn't have to be so embarrassing!)


Funkhouser, Kazhdan, Shilane, Min, Kiefer, Tal, Rusinkiewicz, and Dobkin 2004

Chair Modeled by Merging of Examples Retrieved from a 3D Database

The system described at SIGGRAPH has no notion of material properties or texture, cannot yet work with animated models, and lacks the traditional tools that will still be needed in this brave new world for tweaking smaller areas. But even though search-based editing isn't ready to appear in the next version of your favorite modeling software, it is clearly a compelling alternative that should be further developed. For in-house tools, the ideas of a smart 3D lasso and smooth melding of adjacent shapes are appropriate now, and should be on artists' wish lists.

Beyond the Modeling by Example paper, the SIGGRAPH Proceedings contains two other papers likely to be useful when implementing these ideas. The energy-minimizing curves needed for the smart lasso idea are further explored in Energy-Minimizing Splines in Manifolds. Mesh Editing With Poisson-Based Gradient Field Manipulation is an alternative method for creating the water-tight seams.

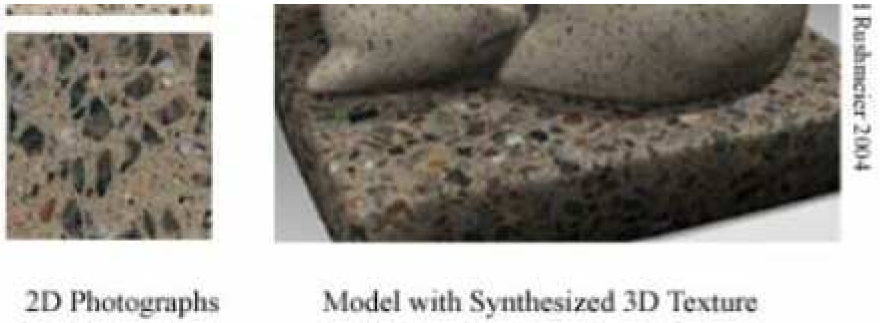
Elsewhere on the modeling front, NURBS are a popular modeling primitive for creating curved surfaces from a set of 3D control points. They appear in many modeling tools. Last year, Sederberg et al. introduced the T-Spline generalization of NURBS that can specify the same surfaces with only a third as many control points. This year, T-Spline Simplification and Local Refinement shows how to convert an existing NURBS model, thus eliminating many control points, and how to locally refine T-splines so that artists can selectively add detail without excessive control points and without cracks in the model. Modeling programs and level editors that incorporate these new T-splines should allow artists to create the same models with less effort.

Textures

A number of papers introduce tools for better accomplishing the texture editing tasks for which PhotoShop's Magic Wand and Clone Brush are typically employed. Interactive Digital Photo Montage seamlessly stitches together multiple images using a few casual mouse strokes. Lazy Snapping, GrabCut, and Poisson Matting use radically different methods from one another to achieve the same results. In each case, an object can be cleanly clipped from its background, including fractional alpha values along the edge, by dragging a box and making a handful of mouse strokes. Compared to the current process of carefully selecting objects with the magnetic lasso and Magic Wand and then manually cleaning edges, these new methods appear painless and make image compositing fun again.

One challenge for both modeling objects that are cut from blocks of material, like statues and caves, and simulating breakable objects in games, is that texture is only skin deep because it is painted on the surface. 3D textures are now supported by graphics hardware and can solve this problem. But how can artists create 3D textures? For 2D textures, we take photographs of real materials and use functions to simulate noisy patterns like spots and stripes. The new Volumetric Illustration method simulates plausible texture for 3D cross-sections given example photographs of real cross-sections. In one example from the paper, the authors use a single photograph of a steak to simulate internal texture throughout an animal, complete with fat striations and different layers of muscle tissue. Stereological Techniques for Solid Textures is an alternative method for materials like rock that contain oddly sized and colored particles within a substrate. This method takes a cross-section photograph, measures the statistical shape and distribution of particles and then synthesizes a 3D volume.

Jagnow, Dorsey, anc



l Rushmeier 2004

Stereological Techniques for Solid Textures creates 3D noise textures from 2D photographs.

Animation

Four papers describe new methods for realistic human motions for both real-time and pre-computed animations relevant to games. Speaking with Hands uses pre-processed speech snippets and animations to synthesize new, synchronized animations and speech at run time. They demonstrated Zoe from Electronic Arts' *SSX 3* giving animated feedback to the player based on their specific actions. The result was believable and as natural as any 'hip and cool' teenage snowboarder can be.

Stone, DeCarlo, Oh, Rodriguez, Stere, Lees, and Bregler 2004

***SSX 3*'s Zoe says, "That was ugly, dude! On this run you forgot to set up your jump."**

Elsewhere, Synthesizing Physically Realistic Human Motion in Low-Dimensional Behavior Specific Spaces presents a new method for optimizing human motion as an offline process. It's something you can imagine incorporated into Character Studio for editing motion capture of a walk cycle to realistically incorporate other motions like jumping and crouching.

Synthesizing Animations of Human Manipulation Tasks is another interesting technique for offline generation of canned animations. It combines AI with animation to create whole animations of characters solving simple physical tasks, like placing a large box in the trunk of a car. The results look great for the simple cases shown in the paper. The character balances her weight appropriately, avoids collisions, and minimizes the energy needed for a task.

Poses created with inverse kinematics, either in real-time or by an artist, obey joint limits but look rather uncomfortable because they don't understand the human body. Style-Based Inverse Kinematics is a new approach to IK that uses a learned model of human poses to produce poses that are likely, instead of just physically possible. Once the data has been learned the method executes in real-time and could be integrated directly into game physics or offline animation packages. From a game programmer's perspective, the drawback of learning-based algorithms is that they are useless without the original data set. However, it would be extremely helpful in implementing this method if the learned parameters for a variety of human figures were made publicly available.

Physics Simulation

Real-time physical simulation has become a must-have feature for new games. It is an area in which we have historically been told that it takes an expert to stabilize numerical integration, and been advised to stick with middleware solutions like Havok. Last year, Guendelman et al. showed that plausible simulation is still a young field where simple insights can make suddenly change all the rules. They showed that inverting the typical order of operations in simulation can make it both easier to implement and extremely stable. This year offers two new ideas that might have similar impact.

Carlson, Mucha, and Turk 2004

Rigid-Fluid's Two-Way Coupling: Falling Objects Splash and Float Animation

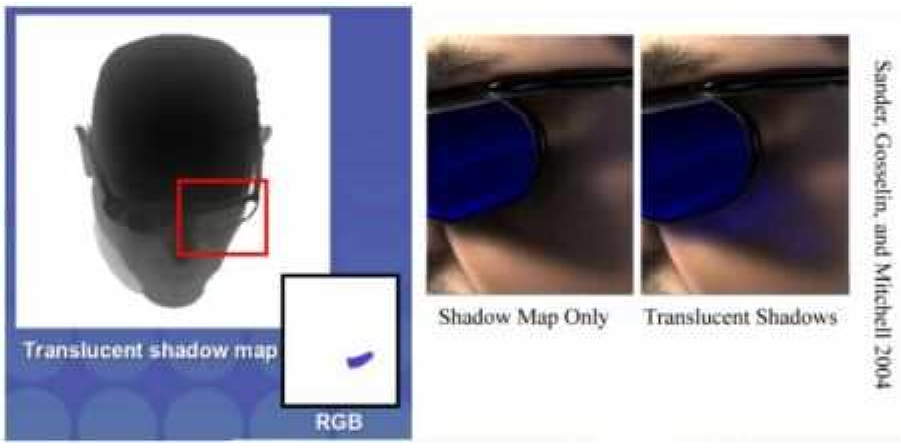
Rigid Fluid simulation solves the Navier-Stokes fluid equations over *everything* in the scene including solids, and then enforces rigid body constraints. This allows fluids to move rigid bodies, and the bodies to push back for the first time. Although the results presented in the paper are not real-time, this method could probably be adapted for game simulation using a coarse fluid grid and GPU simulation techniques. Check out the pre-SIGGRAPH General Purpose Computing on Graphics Processors conference to see just how well GPUs can simulate fluids and particle systems.

The authors of A Virtual Node Algorithm for Changing Mesh Topology During Simulation solve a different problem: allowing deformation in fracture when simulating with coarser meshes than those used for rendering. They achieve this by embedding the mesh in a tetrahedral grid and duplicating instead of subdividing tetrahedra upon fracture. Simulation in games often uses such low-resolution meshes, and, combined with 3D textures, this approach may be useful for creating breakable environments.

Hardware Talks

In addition to the pure graphics research discussed at SIGGRAPH, ATI and NVIDIA each gave a number of presentations on how to get the most from graphics hardware. These combined material previously seen at the Game Developer's Conference with a few new nuggets and information on their new shader analysis suites.

One particularly valuable new idea from these was ATI's transparent shadow map method from the *Ruby: The Double Cross* demo. Shadow algorithms generally assume a single depth is visible for each pixel and don't work well with transparent objects because they violate that assumption. The Ruby team proposes a new kind of shadow map that stores distance from the light source to an opaque surface in the alpha channel and the blended RGB values from transparent objects between the light and that surface in the color channels. The color channel is white where there are transparent objects in front of the light. As with regular shadow mapping, the pixel shader for the visible scene provides no illumination when a surface is farther from the light than the shadow caster. The new twist is that the illumination at points closer to the light is modulated by the RGB value in the shadow map. This makes the transparent surface appear darker than it should but casts correct translucent shadows on opaque objects. When there are multiple transparent surfaces they are colored incorrectly but the error is likely to go unnoticed in many cases.



ATI's Translucent Shadow Map Tracks Both Transparency and Depth

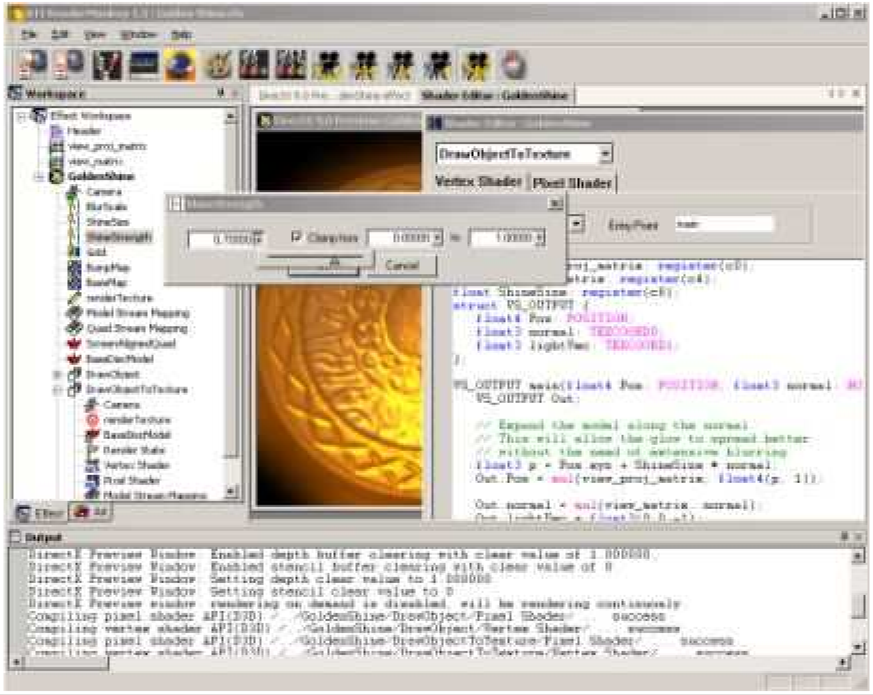
DirectX and OpenGL

The state of the DirectX and OpenGL APIs were also discussed in some detail at SIGGRAPH. Microsoft is following Apple's lead and moving all 2D windowing and rendering to 3D acceleration in the new Windows Graphics Foundation (WGF). This will be the end of the GDI interface and a significant new step for making DirectX both more powerful and more reliable, since many functions will be moved out of the kernel. This also means that DirectX will begin to require multi-tasking at the GPU level since multiple applications will be rendering simultaneously.

Like OpenGL, the new DirectX will also require implementation of a core feature set -- no more capabilities bits and multiple render paths. The programmable GPU model is being extended in several ways. First, pixel and vertex shaders will operate under a unified instruction set model. Dual vertex shaders will allow programmable access to vertices both before and after curve tessellation, and a new Geometry shader will be introduced. The Geometry shader receives an entire post-vertex shader triangle and may alter it in unconstrained ways, including subdivision and killing the entire triangle. These new features make several pieces of the fixed function pipeline obsolete and Microsoft is taking this opportunity to clean house. Expect dinosaurs like triangle fans, point sprites, vertex lighting, and fog to go away soon. Newer cards don't have dedicated silicon for these anyway; they emulate them with the programmable pipeline.

OpenGL is evolving more slowly, although we can probably expect vendors to release GL extensions for the geometry shader and other features before WGF actually ships. The new OpenGL functionality available this summer is mostly catch-up work against DirectX 9.0.

The ARB_texture_non_power_of_two extension allows normal [0, 1] indexing and MIP-maps for textures with arbitrary dimensions. ARB_mirrored_repeat is a new wrapping mode that provides four-fold symmetry for textures and ATI_texture_float is a straightforward floating-point texture format. Programmers hate the convoluted, platform-specific p-buffer APIs needed for rendering to texture under OpenGL (John Carmack has even said that he almost switched to DirectX just to get away from p-buffers on *Doom 3*). The new EXT_framebuffer_object and EXT_pixel_buffer_object provide long over-due relief in the form of a vendor and platform independent off-screen rendering API. One use for p-buffers is shadow map rendering, which is getting another a big boost as well. OpenGL shadow maps now perform percentage-closer filtering automatically for nicely filtered soft shadows and both ATI and NVIDIA's SM3 cards provide depth-only shadow rendering at *twice* the speed of visible surface rendering.



```
...proj_matrix : register(c0);
...atrix : register(c4);
float ShineSize : register(c8);
struct VS_OUTPUT {
   float4 Pos : POSITION;
   float3 normal : TEXCOORD0;
   float3 lightVec : TEXCOORD1;
};

VS_OUTPUT main(float4 Pos : POSITION, float3 normal : N
   VS_OUTPUT Out;

   // Expand the model along the normal
   // This will allow the glow to spread better
   // without the need of extensive blurring
   float3 p = Pos.xyz + ShineSize * normal;
   Out.Pos = mul(view_proj_matrix, float4(p, 1));

   Out.normal = mul(view_matrix, normal);
```

```
DirectX Preview Window: Enabled depth buffer clearing with clear value of 1.000000
DirectX Preview Window: Enabled stencil buffer clearing with clear value of 0
DirectX Preview Window: Setting depth clear value to 1.000000
DirectX Preview Window: Setting stencil clear value to 0
DirectX Preview window: rendering on demand is disabled, will be rendering continuously
Compiling pixel shader API(D3D) ../GoldenShine/DrawObject/Pixel Shader/ ... success
Compiling vertex shader API(D3D) ../GoldenShine/DrawObject/Vertex Shader/ ... success
Compiling pixel shader API(D3D) ../GoldenShine/DrawObjectToTexture/Pixel Shader/ ... success
```

ATI's RenderMonkey Shader IDE

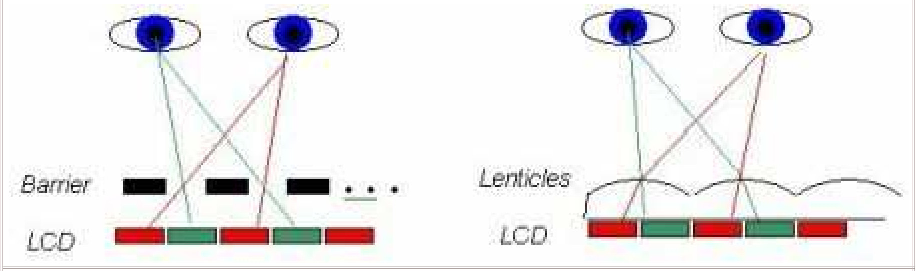
With regard to OpenGL extensions, NVIDIA is to be commended for supporting a number of "ATI_" extensions in their newest drivers and pushing a number of new features into OpenGL as "EXT_" extensions. Vendors have traditionally proposed competing but equivalent OpenGL extensions that just force OpenGL developers to implement features twice. Since extensions are the key that unlocks the advanced features of new graphics cards and consumers don't see the extension names, agreement on common extensions is good for everyone. Developers are more likely to use features that require only one code path. Vendors sell hardware based on the number and quality of features they provide, not behind the scenes IP issues. In the end, consumers want fast cards and good games, and common extensions help produce them.

Although nothing has been formally announced for the APIs, expect to see some new image formats available in both DirectX and OpenGL soon. Likely candidates are OpenEXR for high dynamic range and ATI's 3Dc format for 4:1 compressed normal maps.

3D Displays

Finally, the formerly novelty area of 3D displays got some serious attention at SIGGRAPH. Remember when R2-D2 projects a holographic image of Princess Leia at the beginning of *Star Wars*? The image can be seen (in the world of the movie) simultaneously by multiple viewers, without special glasses, and appears in front of the physical display. While the underlying technology is different from what Lucas envisioned, the experience of today's 3D displays is remarkably similar to what was once only science fiction. A variety of 3D displays are about to hit the consumer and professional markets.

Game developers need to be aware of them for three reasons. Artists may benefit from the enhanced shape perception provided by these displays, which already support popular tools like 3D Studio Max and Maya. Arcades and handhelds have faster technology turnover than PCs and consoles, so developers should immediately consider supporting 3D displays that may be deployed in those markets as early as next year. Most of these displays can be configured to create true 3D images from applications intended only for traditional displays. They do this either by intercepting all rendering calls and executing them multiple times with slightly different projection matrices or by reading the depth buffer to determine the 3D location of each pixel. Developers can easily future-proof their titles without explicitly supporting 3D displays by ensuring that they render scenes in ways that are compatible with these tricks. NVIDIA has added an entire chapter to their GPU Programming Guide on supporting these stereoscopic retrofitting methods.



Directing Different Pixels to Each Eye. (a) Parallax Barrier (b) Lenticular

To make their output appear to actually inhabit 3D space, most of the new displays use a traditional LCD monitor behind either a parallax barrier or a lenticular screen. For each, the application renders stereo views. These are then horizontally interlaced so that odd columns of pixels correspond to the left eye and even ones to the right eye. A parallax barrier is like a fence with one-pixel gaps between the fence posts, set slightly in front of the LCD screen. When the viewer's head is within the display's "sweet spot", only the odd columns are visible through the gaps to the left eye; the posts block the even columns completely. The opposite is true for the right eye. The viewer thus sees two separate images as if they were wearing red/green or shutter glasses, but the image is in true color and at the full refresh rate of the display. A lenticular screen instead directs different columns to each eye using one-pixel wide cylindrical lenses. You've seen the effect before as a gimmick on airport billboards, stickers, and CD covers. In those cases, instead of 3D views, they often display separate frames of an animation. Two views gives a small sweet spot. If the viewer moves more than a few centimeters the effect breaks down and may appear to have inverted depth, moiré, and diffraction patterns. To make the sweet spot large enough for a single viewer to move comfortably and for multiple people to share a display, four, eight or sixteen views can be rendered. This also creates a parallax effect so that by moving their head, the viewer can look behind objects. Because they divide the pixels of a single monitor over multiple views, both barrier and lenticular displays have intensity and resolution divided by the number of views, which makes them dim and pixilated compared to 2D displays.

Sharp was the first to reach consumers with a mass market 3D display -- their two-view barrier display cell phone is already available in Japan and the 15-inch LL-151-3D LCD will soon be appearing on laptops and PC's. Sanyo's TM080SV-3D is an 8-inch LCD barrier display with four views, each at 600x200 resolution. It is much brighter and sharper than Sharp's entry. Viewing cartoony games that appear to stand out of the tiny display immediately immediately makes one think of Nintendo-hopefully the next handheld after the DS will have a 3D display as good as this one.

X3D takes the displays to the next level and is the minimum for what you would want on your desktop. Their eight view barrier displays are now available from 4-inch to 50-inch, with the 17-inch priced at a very affordable $1,500. The company is aware that content availability is the key to launching their displays. They provides the X3D OpenGL Enhancer automatically renders eight views for normal OpenGL programs and Vice President of Content Services Ken Hubbell is actively seeking game developers to establish partnerships and build in native support for the displays. They are also developing technology to automatically convert console games to work with 3D televisions. X3D's displays are bright, sharp, and large (if low resolution), and will be appearing in casino and arcade machines and as digital signage in a few months.

Ampronix Inc. is one of the few companies pursuing lenticular technology. They show two 800x1200 views on their $4,000 20-inch display. Because Ampronix only provides stereo output, it works with any regular stereo card and is a drop-in replacement for shutter glasses. The downside of this display is horrible flicker and diffraction if the viewer's head isn't within 2.5 cm of the ideal position.

Kodak has a unique technology in which viewers look through a spherical lens that merges stereo 1280x1024 images, the highest resolution available on any 3D display. It is targeted at CAD users and carries a price tag in the tens of thousands of dollars. The major drawbacks of this display are the huge footprint (similar to a washing machine) and that it requires the user to place his head against a very small window to precisely match the tiny sweet spot of the unusual lens.

Lightspace Technologies has the DepthCube Z1024, the only display where the viewer actually focuses at different depths, and by far the brightest on the show floor. Their unique technology has a rear projector that illuminates a stack of 20 screens inside a cube. The individual screens can be toggled between opaque and transparent. An extremely high frame rate projector paints separate images at the 20 different depths in the time a conventional display paints a single 2D image. This means that the object seen is really 3D, with parallax, perspective foreshortening, and ocular convergence. Combined with an effectively unlimited sweet spot, this display likely produces less fatigue and a more subtly compelling sense of depth. The downsides are 15-bit color (displaying a real 3D image requires huge bandwidth), CRT-size footprint, and depth shadows. These depth shadows appear as black outlines on objects when viewed off-axis (as if wearing a head lamp). Also, the price is prohibitive at tens of thousands of dollars. Because of this, the company is not considering the consumer or art workstation markets but is focusing on scientific visualization, security, and engineering markets.

SIGGRAPH - Roll on 2005!

Overall, this year's SIGGRAPH was both interesting and thought-provoking. You can download the complete SIGGRAPH Proceedings from the ACM Digital Library or read collected preprints on the web for free.

SIGGRAPH returns to the Los Angeles Convention center July 31, 2005 and will be in Boston in 2006. The papers submission deadline is January 26, 2005 -- consider not just attending but also submitting your work. Although full papers require time and detailed disclosures that are not appropriate for many game developers, sketches, posters, and the animation festival are great venues to advertise your new title and get credit for your innovation.