Matrox G400

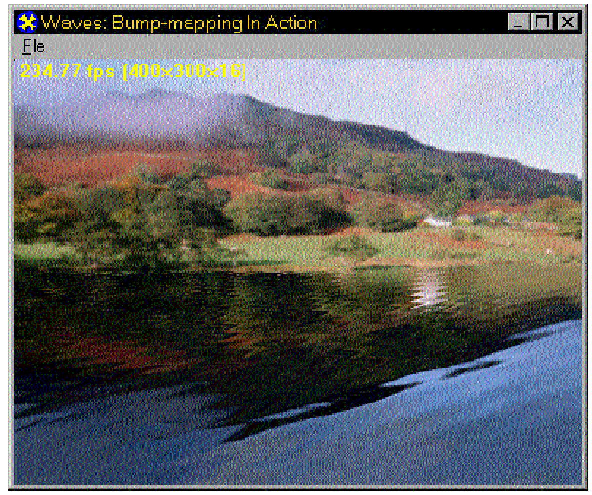
Environment Mapped Bump Mapping

Dramatically enhancing the visual realism of 3D Scenes

Environment Mapped Bump Mapping is a DirectX 6 quality feature which will be used to substantially increase the visual realism of 3D scenes. In fact, 3D graphics hardware that supports this feature will be able to render 3D scenes with more realism than was ever before possible on the PC.

Environment Mapped Bump Mapping is essentially a technique that allows a much higher level of detail to be added to a 3D world than could be possible with texture-mapped polygons alone. Fine details such as the pock-marked surface of bricks in a dungeon and scratches on robots and tanks can be added with ease. Special effects such as realistic water surfaces, heat shimmering off hot asphalt on a summer day and air turbulence in flight simulators can also be uniquely accomplished by using Environment Mapped Bump Mapping. This new feature will prove to be as revolutionary as alpha blending in terms of the creative effects that game developers will accomplish when given free rein. With the advent of the Matrox G400 as the first chip to support Environment Mapped Bump Mapping in hardware, game developers are quickly pledging to add support for this feature into cutting-edge games that will ship in 1999 (see below).

FIGURE 1



Environment Mapped Bump Mapping on the Matrox G400 can be used to simulate the movement of waves.

Environment Mapped Bump Mapping can be used in various ways to add compelling effects to a 3D scene. For example, Environment Mapped Bump Mapping has the capability to create animated bumps, as well as fixed or motionless bumps on the surface of a 3D object. An animated bump can be applied to any 3D object or to any part of a 3D scene to simulate various environmental effects such as waves and ripples on water (figure 1) and distortion in the atmosphere due to heat or turbulence.

Although other chips currently shipping may claim to support bump mapping, they do so using multi-pass alpha-blending capabilities to simulate bumps on textured 3D objects. It is worth noting that this method offers only a fraction of the effects possible with true Environment Mapped Bump Mapping. In addition, multi-pass alpha-blending produces artifacts, is limited to monochrome lighting and is unable to simulate any environmental effects.

Early Content List:

- Slave Zero from Accolade
- Black & White from EA/Lionhead
- Expendable from Rage Software
- Messiah from Interplay/Shiny
- Descent III from Interplay/Outrage
- Draken from Psygnosis/Surreal
- SpeedBusters from Ubisoft
- Rally Masters from Gremlin/DICE
- Midnight GT Racing, Hostile Waters and Incoming Forces from Rage Software
- ShadowMan from Acclaim/Iguana
- Team Alligator and Jump Runner from Simis
- BattleZone 2 from Activision/Pandemic

matrox

Adding bumps to 3D objects

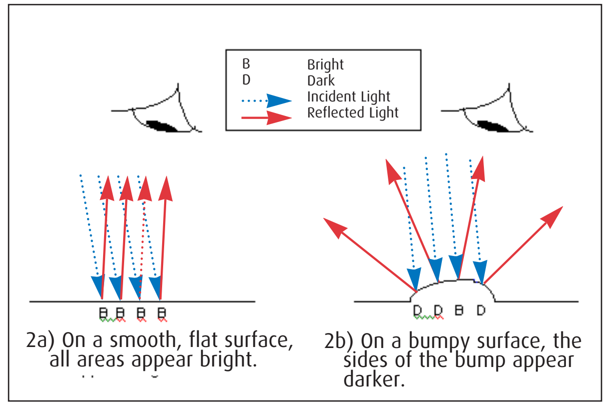
In its simplest form, bump mapping adds realism to textures and objects by creating the illusion of bumps, or variations in surface depth, on an otherwise flat surface. This means that a flat plane can be transformed into the dimpled surface of a golf ball, the gnarled bark of a tree or the rough surface of a rock.

For bump mapping to be convincing, the eye must perceive variations in surface depth even though the surface really is flat. The viewer's perception of depth is influenced by the amount of light reflected into the eyes by the surface being looked at, and the amount of light reflected in any given direction depends on the surface reflecting it. A smooth, level surface will reflect more light than a bumpy one. Likewise, a surface that is perpendicular to the viewer's line of vision will reflect more light than one that is not directly facing the viewer.

Consider the situation in figure 2. A flat surface (2a) directly facing the viewer reflects more light back into the viewer's eyes than the bumpy surface does. The viewer perceives all areas of the surface as being of equal brightness and consequently, completely flat. However, the bumpy surface (2b) reflects less light back into the viewer's eyes because all areas of that surface are not directly facing the viewer. The part of the bump that is directly in the viewer's line of vision appears brighter than the parts that are not--meaning that the sides of the bump appear darker, and may also cast shadows.

In 3D graphics, the illusion of bumps relies solely on color, lighting and shading. The lighting/shading effect is

FIGURE 2

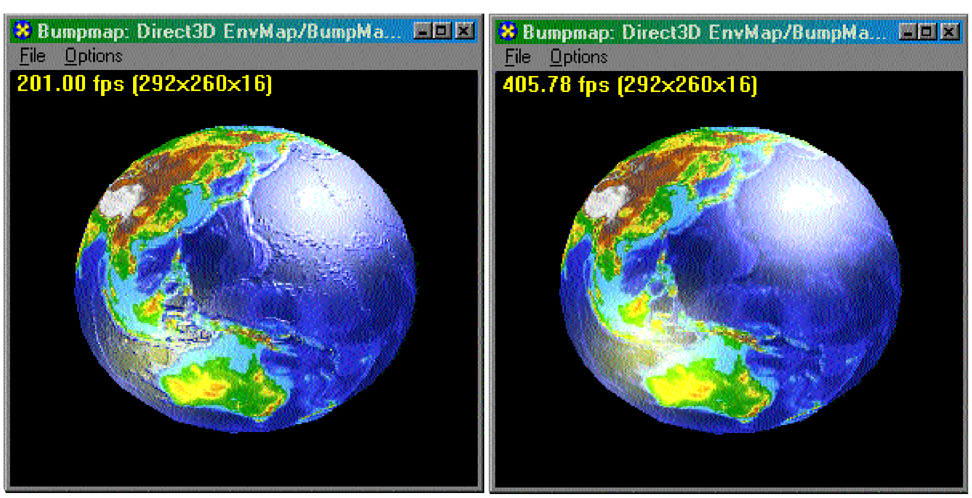


Reflection of light from flat and bumpy surfaces

easily achieved by modifying the color of each of the texels, or textured pixels, being drawn on the screen. Using a predetermined algorithm, the color of each texel is modified to represent the amount of light that would be reflected from any given point on a bumpy surface. This depends on the angle of the light source relative to the angle of the surface at the point of reflection.
In figure 3, the different altitudes of the Earth's surface are made obvious by the variation in lighting. Areas that are brighter are directly facing the light. Other areas are made darker either because they do not face the light source, or because they lie in the shadow of more elevated areas. Either way, the areas that are darker receive less light.

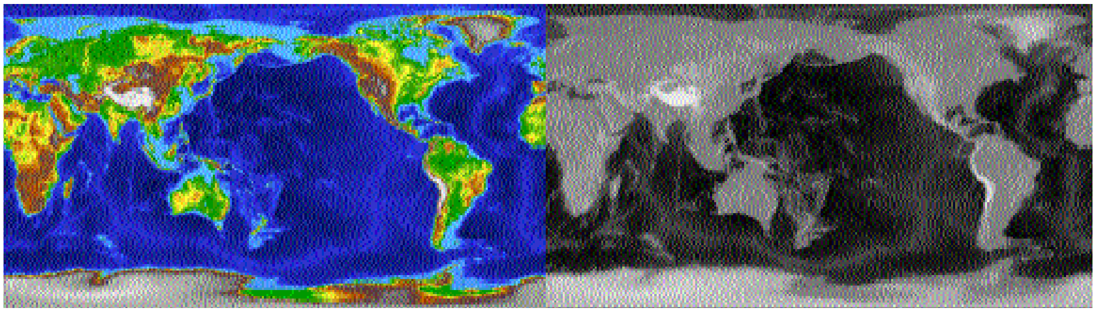
Bump mapping uses lighting and shading effects on the texture to create the appearance of surface depth, but the

FIGURE 3



The same 3D model is shown with (left) and without (right) Environment Mapped Bump Mapping.

FIGURE 4



A conventional texture map | A bump map

variation in surface depth is really just an optical illusion. Figure 3 is a good example of this. Although the surface of the 3D model on the left appears jagged and in relief, its outer edge is not marked by bumps. To create the illusion of bump mapping, the only thing that has been modified is the color of the texture on the surface of the sphere on the right.

To create this effect, all areas of the surface must be specified as either elevated or sunken. To do this, a second texture or "bump map" is required to define depth or height for every pixel in the conventional texture map. In the simplest case, this bump map would be a monochrome intensity map with brighter shades representing the more elevated areas and the darker shades representing the lower areas (figure 4). A gradual change in intensities represents a moderate slope, and a sudden change in intensities represents an abrupt change in depth. Alternatively, the bump map could be called a height map.

To complete the effect, an "environment map" is required. The environment map represents what should be reflected off the surface. It can be as simple as a bright circle representing a single light source or a complex environment map consisting of multiple light sources and effects due to other environmental factors such as clouds and shadows.

Multi-pass alpha-blending and its limitations

Some existing 3D hardware can simulate one bump mapping effect known as ‘embossing’ with multiple alpha-blending passes. However, this is a per-polygon technique which produces artifacts, is limited to monochrome lighting and is unable to simulate any of the more interesting environmental effects. Embossing with multiple alpha-blending passes is accomplished when a monochrome version of the conventional texture map is shifted in one direction, subtracted from the original texture, shifted back, and then blended into the original texture map. This gives the illusion of shadow on one side and bright light on the other. The direction and amount of shift is determined by the direction and intensity of the light source.

Figure 5 shows a comparison between embossing using the limited multi-pass alpha approach (top) and embossing using Environment Mapped Bump Mapping (bottom). Notice how the Environment Mapped Bump is much more realistic than the multi-pass alpha; it is easy to spot that the image on the top is simply a shifted replica of the original map.

FIGURE 5
Embossing rendered using multi-pass alpha-blending. Note the artifacts in this example.

True Environment Mapped Bump Mapping

Although multi-pass alpha can be convincing at times, it has several critical limitations with its implementation and results, as detailed in table 1. Some of these include:

Limited lighting angle

The illusion rapidly breaks down and looks wrong when the light is past a certain angle of incidence (it cannot shift indefinitely). At that point, it becomes obvious that the shadowed and lit areas are simply shifted versions of the actual texture, as shown in figure 6.

FIGURE 6

Light Limitations

Limited range of bumps

Multi-pass alpha is limited in the range of bumps supported. This technique cannot accommodate varying degrees of height as the shift must be uniform over each polygon.

Complex implementation

The proper amount of shifting, or shadow, has to be calculated by the application. This can be tricky to get right, and just means more work for the programmer and the host CPU. The calculation of the amount of shift becomes even more complicated if multiple light sources are present.

It is easy to see that with all of its limitations and disadvantages, this technique lacks ease of use for developers. In fact, it cannot even really be considered a hardware technique because most of the work is done by the application, which means that it uses precious CPU cycles.

Most importantly, multi-pass alpha cannot be used to simulate any of the special effects and animations that Environment Mapped Bump Mapping supports.

Environment Mapped Bump Mapping

Given the limitations of multi-pass alpha, DirectX 6 uses a more versatile and flexible method of implementing bump mapping in hardware. This method is Environment Mapped Bump Mapping.

In typical environment mapped texture mapping (without bump mapping), the environment map contains a value for each texel of the conventional texture map. This value is applied to the texture map when it is mapped onto the 3D surface.

When adding bumps, the process becomes more complex.

Environment Mapped Bump Mapping requires three maps: a conventional texture map, a bump map and an environment map. A bump map is essentially a differential value map derived from a height map (see figure 7). The bump map contains a value for each texel coordinate of the conventional texture map. This value specifies a value from a different location within the environment map that should be applied to the texture map.

The bump map is combined with the environment map and the resulting "perturbed" environment map is applied to the original texture. In other words, for each texel coordinate of a texture map, there exists a corresponding environment map coordinate, the value of which is applied to the texel in order to create the lighting effect.

In figure 7, the environment map is a single-source monochrome light map. If the light source in the environment map is positioned on the left side of the original texture, then the final result looks like figure 7a, which is typical texture mapping.

The Algorithm

The bump mapping high-level algorithm can be specified in the following form:

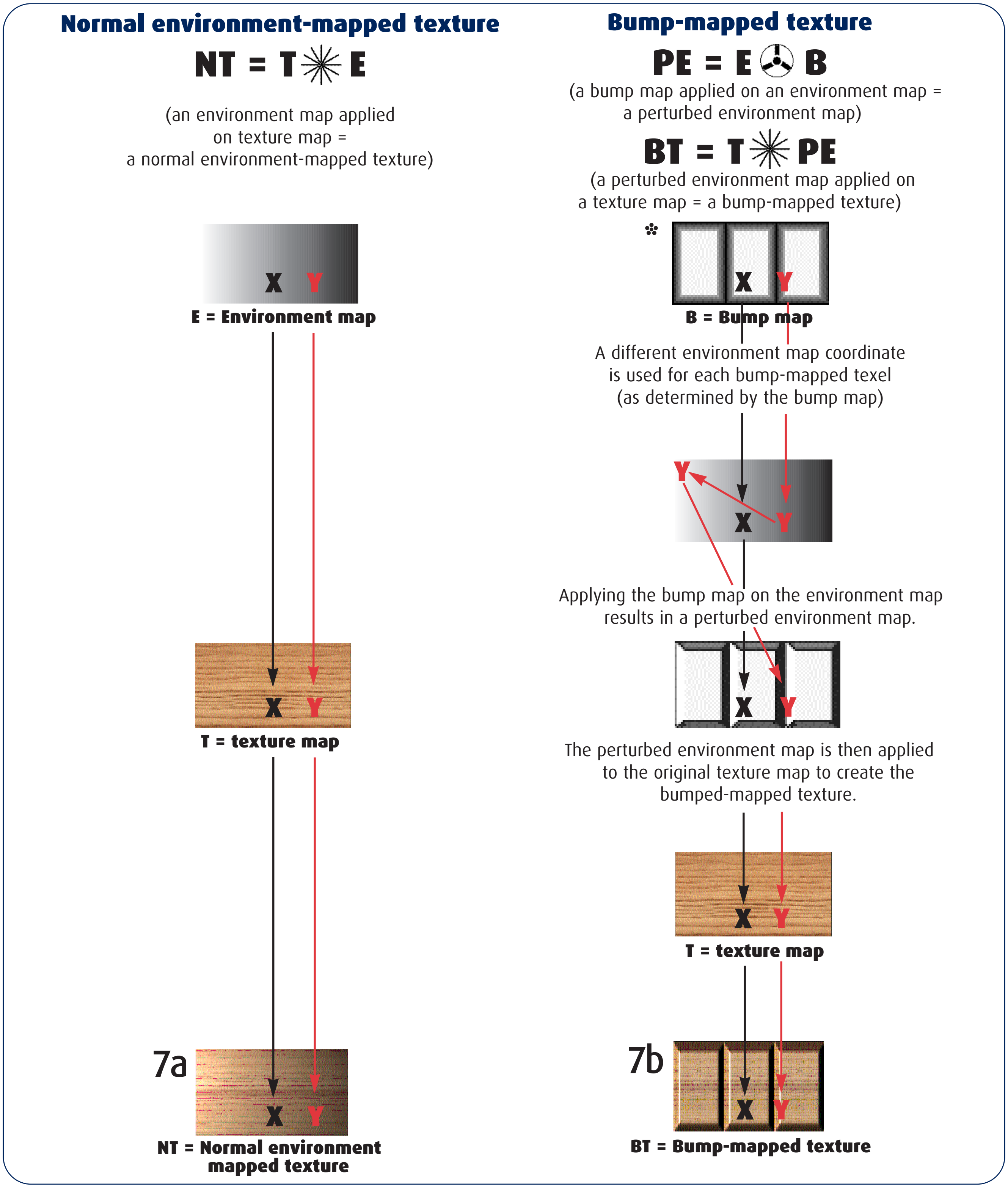
Define:
T = Texture map
E = Environment map
B = Bump map
PE = Perturbed environment map
✳= Environment mapping operation
☢ = Bump mapping operation
NT = Normal environment-mapped texture
BT = Bump-mapped texture

Normal environment-mapped texture
NT = T ✳ E
(environment map applied to a texture map = normal environment mapped texture)

Bump mapped texture
PE = E ☢ B
(bump map applied to a environment map = perturbed environment map)
BT = T ✳ PE
(perturbed environment map applied to a texture map = bump-mapped texture)

Matrox G400

FIGURE 7



Environment Mapped Bump Mapping

* Note that the monochrome bump map texture is used to represent the bump map for clarity. In reality, the bump map is just a series of values for each texel derived from this texture.

Matrox G400

As explained earlier, a bump is simulated by changing the lighting on the areas where the bump introduces a slope on the surface. Therefore a different environment map coordinate is used for the same texel coordinate. This new environment map coordinate is defined by the bump map and is dependent on the slope of the bump at that particular coordinate. Notice how for the same coordinate, the bumped texture chooses a darker or lighter shade from the environment map, giving the impression that the slope at that point is not flat relative to the direction of the light.

Advantages of Environment Mapped Bump Mapping

The advantages of Environment Mapped Bump Mapping as supported by DirectX 6 and the Matrox G400 are presented in more detail in table 1. Highlights listed in the table include:

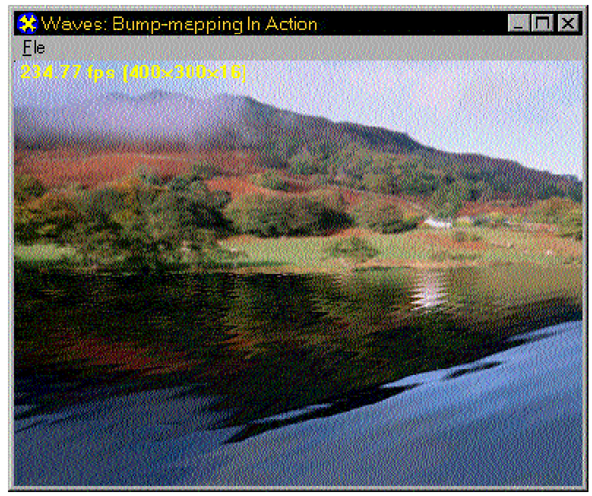
- Environment Mapped Bump Mapping can accommodate multiple light sources in one pass, and reflective environment mapping on the same bump, because all of these additions become part of the environment that is being mapped on the bump.

- In addition, the bump map does not have to change from frame to frame. Unlike the multi-pass alpha technique, Environment Mapped Bump Mapping is per pixel rather than per polygon. This technique is non-restrictive and therefore is much easier for developers to implement.

- Environment Mapped Bump Mapping can also be used to simulate animated bumps such as reflective waves on a water surface, as shown in figure 8. When running this scene, the waves move with exceptional fluidity and realism. Surprisingly, this frame contains only four triangles, and yet the movement of the waves is as smooth as video. This effect is obtained by perturbing or distorting the environment, using a dynamic or animated bump map while applying it to the texture.

The effect in figure 8 is produced using a procedural approach to bump mapping. The water texture is created procedurally using a dynamic bump map (wave effect) along with the reflective environment (the texture for the mountain). There is no pre-assigned texture for the water in this case. In other words, bump mapping can be applied to a procedurally-created texture for a wide range of compelling effects.

FIGURE 8



Environment Mapped Bump Mapping on the Matrox G400 can be used to simulate the movement of waves.

TABLE 1

Multi-pass alpha	Environment Mapped Bump Mapping
· Multi-pass alpha is limited to only simulating bumps on textured 3D objects. Procedural bump mapping or environmental bump mapping cannot be simulated.	· Environment Mapped Bump Mapping can simulate static, dynamic and procedural bump mapping. Furthermore, these bumps can be applied to textured 3D objects or whole environments.
· Multi-pass alpha cannot be used for distortion effects on the environment as this technique cannot use an environment map. Therefore, effects such as waves and turbulence cannot be simulated.	· Since Environment Mapped Bump Mapping uses an environment map, it can use the bump map to simulate distortion effects on the environment. Figure 8 shows one example of how this can be used to simulate waves with full reflection effects. The type of effects that can be simulated with Environment Mapped Bump Mapping are limited only by the creativity and imagination of the developer as this technique offers enough flexibility to be used in numerous ways.
· Since different hardware supports different techniques for multi-pass alpha, results will vary significantly depending on the hardware. Images will likely look different from what the application developer intended.	· Since this feature is part of a standard API, the resulting images will always look similar to what the application developer intended on all supporting hardware. The only differences will result from how well the feature is implemented (quality).
· Since different hardware supports different techniques for multi-pass alpha, the application developer needs to support and/or tweak for all these different techniques. This includes tweaking of both artwork and code-path for all cases.	· Application developers program bumps using the standard DirectX 6 calls, independent of hardware.
· Range of bumps is restricted due to hardware limitations (e.g. some hardware does not support subtraction).	· Range of bumps is largely unlimited.
· UV shifts that simulate the bump are calculated on a per-vertex basis; therefore, the height/depth of the bump can vary only linearly over the triangle surface.	· Bumps are calculated on a per-pixel basis. Therefore, the height/depth of the bump can vary over a triangle and thereby result in a more realistic image.
· The CPU calculates UV shifts. Therefore, multi-pass alpha is CPU dependent.	· All the per-pixel calculations are done on the graphics hardware.
· For the multi-pass alpha technique to look acceptable, the UV shifts would have to be done on many points on the surface to approximate the right shift (direction of shadow) that a light would cast from each specified position. For this to happen, the models would have to have a lot of vertices, thereby increasing the complexity of the artwork and geometry, and reducing performance.	· Since the bump map is applied to the surface on a per-pixel basis, the technique looks good regardless of the number of vertices. Furthermore, each calculation is done in hardware so the load on the CPU is minimal.

(continued)

Matrox G400

TABLE 1 (Continued)

Multi-pass alpha	Environment Mapped Bump Mapping
· The technique breaks down quickly when the light hits a vertex from a wide angle. This is because the UVs can be shifted only so much without disrupting the illusion (figure 6).	· Since Environment Mapped Bump Mapping does not use shifts to simulate bumps, there are no such limitations.
· Since shifts are calculated based on the relation of the light source to the target surface, the bump illusion would not be visible if the light source was perpendicular to the surface (the UV shift would be 0).	· Environment Mapped Bump Mapping does not rely on shifts and therefore, there is no such limitation.
· Multi-pass alpha uses diffuse lighting only.	· Specular lighting effects are available with Environment Mapped Bump Mapping.
· With the multi-pass alpha technique, only one light source can be supported. Furthermore, this source must be monochrome. To accommodate additional light sources, extra passes per light source are required.	• Environment Mapped Bumped Mapping uses an environment map to calculate the bump. This environment map can contain multiple light sources without requiring any additional passes. Furthermore, the light sources can be polychromatic.