

VISUAL DISTORTION EFFECTS THROUGH TRANSLUCENT STRUCTURES IN DESIGN SOFTWARE

BACKGROUND OF THE INVENTION

5 The Field of the Invention

The present invention relates generally to computer-aided design or drafting software.

Background and Relevant Technology

Computer systems now play a critical role in building and/or architectural design. In particular, architects and interior designers (“or designers”) use a wide range of computer-aided design (CAD) software for designing the aesthetic as well as functional aspects of a given residential or commercial space. For example, a designer might use an architectural design program (e.g., a CAD program) to design the interior layout of an office building. The designer might then render the layout to create a three-dimensional model of the interior of the office building that can be displayed to a client.

15 Some CAD programs may provide the designer with the ability to create three-dimensional models using a variety of different building materials. For example, the designer may be able to design a particular entryway within the three-dimensional model to have marble floors and wood paneling on the walls. Similarly, the designer may design an office space to have a large window looking outside and a frosted glass window inset within an office door. One will understand that allowing a designer to utilize a variety of materials when designing a three-dimensional model will result in a more visually appealing three-dimensional model. Additionally, allowing a designer to utilize a variety of materials when designing the three-dimensional model will allow the model to more accurately reflect a real-world building.

25 In at least some situations, a designer may choose to use a glass panel as a wall within the three-dimensional model. For example, when designing a restroom shower, a designer may choose to create the shower stall with a glass panel as a door. Similarly, a designer may choose to use a glass panel within a three-dimensional model of an office. Specifically, the designer may use the glass panel as a part of a door to the office or as a decorative feature positioned as a wall.

30

Some conventional CAD programs may provide a designer with the ability to incorporate glass panels into a three-dimensional model. In some cases, however, a designer may desire to incorporate a translucent panel into a three-dimensional model. For example, a designer may desire to create a decorative wall portion that is constructed of a translucent glass (e.g., a frosted glass panel inset within a door). Unfortunately, certain types of effects for viewing translucent surfaces within a three-dimensional model tend to be very limited.

For example, many translucent surfaces allow only distorted images and distorted light to pass through the surface. Many conventional CAD programs, however, have only limited, if any, ability to distort the objects on the opposing side of the translucent surface in a realistic manner. In some cases, conventional CAD programs may completely neglect to distort the image in any way. In contrast, some conventional CAD programs may render the surface such that there is essentially no indication of any objects on the opposing side of the translucent surface. One will understand that providing a realistic distortion effect caused by translucent surfaces can be beneficial to designers using CAD programs because it will allow a designer to visualize and understand in a more realistic way the three-dimensional models that the designer has drawn.

Accordingly, an advantage in the art can be realized with systems, methods, and computer program products that distort images that are at least partially viewable through translucent surfaces.

BRIEF SUMMARY OF THE INVENTION

Implementations of the present invention overcome one or more problems in the art with systems, methods, and apparatus configured to create translucent surfaces within a three-dimensional model. In particular, at least one implementation of the present invention renders a translucent surface that distorts a view of objects on an opposing side of the translucent surface. Additionally, in at least one implementation of the present invention, a distortion effect rendered with respect to a translucent surface depends upon the composition of the surface. Furthermore, at least one implementation of the present invention allows a user to create realistic visual distortion effects through translucent structures within a three-dimensional model.

For example, a method in accordance with at least one implementation for creating translucent surfaces within a three-dimensional model can include rendering a three-

dimensional model of a layout of a design space within an architectural design environment. The method can also include identifying a first portion of the three-dimensional model that is positioned behind a translucent surface within the three-dimensional model. In addition, the method can include rendering a visual distortion effect for the identified first portion of the three-dimensional model. Furthermore, the method can include displaying on a display device the three-dimensional model comprising the translucent surface. As such, the three-dimensional model is displayed behind the translucent surface in a distorted form based on the rendered distortion effect.

In an additional or alternative implementation, a method for rendering a translucent surface within a three-dimensional model can include rendering a three-dimensional model of a layout of a design space within an architectural design environment. The method can also include identifying a translucent surface within the three-dimensional model. Additionally, the method can include performing a distortion effect on the translucent surface. In such a case, the distortion effect can include masking a portion of the three-dimensional model from the translucent surface. Additionally, the distortion effect can include blurring one or more of the pixels of the translucent surface. Further, the method can also include displaying to a user the three-dimensional model, including the translucent surface.

These and other objects and features of the present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

In order to describe the manner in which the above recited and other advantages and features of the invention can be obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. It should be noted that the figures are not drawn to scale, and that elements of similar structure or function are generally represented by like reference numerals for illustrative purposes throughout the figures. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

Figure 1 illustrates an architectural schematic diagram of a system for creating a translucent surface within a three-dimensional model;

Figure 2 illustrates a translucent surface within a three-dimensional model;

Figure 3 illustrates a stencil buffer comprising at least a portion of the translucent surface;

Figure 4 illustrates a rendered scene and stencil buffer;

Figure 5A illustrates a translucent surface prior to the application of a distortion effect;

Figure 5B illustrates a translucent surface after the application of the distortion effect;

Figure 6 illustrates a flow chart of a series of acts in a method in accordance with an implementation of the present invention for creating a translucent surface within a three-dimensional model; and

Figure 7 illustrates another flow chart of a series of acts in a method in accordance with an implementation of the present invention for creating a translucent surface within a three-dimensional model.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Implementations of the present invention extend to systems, methods, and apparatus configured to create translucent surfaces within a three-dimensional model. In particular, at least one implementation of the present invention renders a translucent surface that distorts a view of objects on an opposing side of the translucent surface. Additionally, in at least one implementation of the present invention, a distortion effect rendered with respect to a translucent surface depends upon the composition of the surface. Furthermore, at least one implementation of the present invention allows a user to create realistic visual distortion effects through translucent structures within a three-dimensional model.

For example, at least one implementation relates to displaying, in real-time, a three-dimensional model that comprises a translucent surface. As used within this application a translucent surface can comprise any surface that either distorts or partially blocks at least some light as it passes through the surface. A translucent surface can comprise glass, plastic, or any other material through which at least some light can pass.

Additionally, in at least one embodiment, a translucent surface can provide a distorted view of objects that are on an opposing side of the surface from the user's perspective. For example, a three-dimensional model of an office may include a wall that is constructed, at

least in part, of a frosted glass panel. The frosted panel may separate the office from a sitting area. In at least one implementation, the software provides a distorted view of the sitting area to a user as viewed through the frosted glass panel within the office.

5 In addition, in at least one implementation the distortion effect that is created by the translucent surface depends upon the composition of the translucent surface. For example, the translucent surface may only be slightly frosted such that an object on the opposing side of the translucent surface is only slightly distorted. Similarly, the translucent surface can comprise a colored or a tinted material such that the software displays an object on the opposing side of the translucent surface as distorted and slightly colored with respect to the color of the
10 translucent material.

Further, in at least one implementation a mask can be used when distorting the translucent surface to prevent artifacts from appearing within the distorted surface. For example, a mask can be used to prevent the distortion algorithm from sampling elements of the scene that are bordering the translucent surface. As used within this application, a “mask”
15 comprises any construct meant to temporarily prevent a specific portion of the scene from being accessed. For instance, in at least one implementation, a mask comprises rendering the entire scene, except the area to be distorted, as a single color, such as black. As used in this implementation, whenever the distortion algorithm accesses an area of the scene that is black, that area is not used in computing a distortion effect. One will understand that a mask can be
20 created in a variety of different ways and still remain within the scope of the present invention. For example in at least one implementation, a mask can be created by a color other than black.

Accordingly, one will appreciate in view of the specification and claims herein that at least one implementation of the present invention allows a user to create a three-dimensional
25 model comprising translucent surfaces. The inventive software can accurately and efficiently display the model through a graphical user interface just as the parts would look in real-life, after assembly. Specifically, at least one implementation of the present invention allows a user to create a translucent surface that at least partially distorts the view of an object on an opposing side of the translucent surface. In addition, at least one implementation further
30 allows for different distortion effect depending upon the make-up of a particular translucent surface.

Along these lines, Figure 1 depicts an architectural schematic diagram of a computer system 130 for creating a translucent surface within a three-dimensional model 105. In particular, Figure 1 shows user input devices 110 that communicate with the computer system 130, which in turn communicates with a display 100. Figure 1 shows that the user input devices 110 can include any number of input devices or mechanisms, including but not limited to a mouse 112a, a keyboard 112b, or other forms of user input devices, including any number of remote or touch-screen oriented input devices.

In addition, Figure 1 shows that computer system 130 comprises a design software application 140 executed by a processing unit 150. One will appreciate that the processing unit 150 can comprise a central processing unit, a graphics processor, a physics processor, or any other type of processing unit. Figure 1 further shows that the computer system 130 can comprise a storage device 160. In at least one implementation, the storage device 160 can comprise RAM, a hard disk, solid state memory, or any other applicable computer storage medium. The storage device 160 can contain, among other things, templates and objects that can be placed within a three-dimensional model 105. Additionally, the storage device can be used for temporary storage of frequently accessed items. These components, in conjunction with processing unit 150, store and execute the instructions of design software application 140.

Figure 1 shows that a user in this case uses the input device(s) 110 to send one or more request 120a to the computer system 130. In one implementation, the processing unit 150 implements and/or executes the requests from user input devices 110 and design software application 140 instructions. For example, a user can provide one or more inputs 120a relating to the design and rendering of a three-dimensional model 105 within a design software application 140, as executed by processing unit 150. Figure 1 further shows that the design software application 140 can then pass the inputs 120b to the appropriate modules within the design software application 140.

Ultimately, design software application 140 can then send corresponding rendering instructions 120c through the rendering module 145b to the display 100. As shown in Figure 1, for example, the display 100 displays a graphical user interface in which the user is able to interact (e.g., using the user input devices 110, or perhaps directly in the case of a touch screen display). In particular, Figure 1 shows that the graphical user interface can include a

depiction of a three-dimensional model 105 of a design space, comprising in this case two items of furniture separated by a clear glass panel and a frosted glass panel.

One will appreciate in view of the specification and claims herein that the user interface module 135 can provide to the user an option to make design changes to the three-dimensional model 105. In at least one implementation, for example, upon receiving a request for some modification, the user interface module 135 can communicate the request to the design module 145a. One will appreciate that the design module 145a can then provide the user with the ability to, among other options, place new objects within the three-dimensional model 105, manipulate and change objects that are already within the three-dimensional model 105, place translucent surfaces within the three-dimensional model, or change parameters relating to the three-dimensional model 105. In some cases, this can include the design module 145a communicating with the storage device 160 to retrieve, among other things, templates and objects that can be placed within a three-dimensional model 105.

After receiving and processing a user input/request, the design module 145a can then send a corresponding request to the rendering module 145b for further processing. In one implementation, for example, this further processing can include rendering module 145b rendering the depiction of the three-dimensional model 105 shown, in this case, on the display 100 in Figure 1. One will appreciate that the rendered depiction can include a translucent surface.

Figure 1 further shows that the rendering module 145b can communicate with a stencil buffer submodule 145c. In at least one implementation, the stencil buffer submodule 145c can identify an area where a translucent surface has been placed within a three-dimensional model 105, and load that area into a stencil buffer. Thus, in at least one implementation, when the stencil buffer submodule 145c loads the area where the translucent surface has been placed into the stencil buffer, the area (behind the translucent surface) is not initially rendered. Rather, the rendering module 145b renders the rest of the scene that is outside the translucent surface, and then subsequently renders that area behind the translucent surface, as discussed more fully herein.

One will appreciate that the use of a stencil buffer, in this case, is merely exemplary. In general, a stencil buffer can be used to create a mask over a certain portion of a scene. The

scene can then be rendered while excluding the portion that is covered by the mask from rendering. One will understand, however, that other methods are known in the art for creating a mask over a portion of a scene, such that the masked portion is not rendered.

Additionally, Figure 1 shows that the rendering module 145b can communicate with a mask submodule 145d. In at least one implementation, the mask submodule 145d can create a mask over a portion of the scene to prevent the masked portion from being accessed by a distortion algorithm. (Otherwise, the software applies the distortion algorithm to all unmasked surfaces within the three-dimensional model 105 – e.g., those behind the translucent surface, or elsewhere as appropriate). For example, in at least one implementation, the mask submodule 145d can render a portion of the scene as a single color. When applying a distortion algorithm to a scene, the rendering module 145b can then ignore any portion of the scene that is the same color as the mask. As will be described at greater length below, using a mask in this way may help prevent the appearance of artifacts within a translucent surface in the three-dimensional model 105.

One will understand that there are several different methods for creating a mask. For example, a mask can be created by rendering a particular area to comprise a distinct characteristic, such that it is apparent to the rendering module 145b that the area has been masked. Additionally, in at least one implementation, an alpha channel can be utilized to create a mask.

Similar to above, one will understand that in at least one implementation the stencil buffer can be used to create a mask over the scene. In this application, for the sake of clarity, the stencil buffer submodule 145c is referenced with respect to the translucent surface and the masking submodule 145d is referenced with respect to the rest of the scene. One will appreciate, however, that in at least one implementation, the masking submodule 145d and the stencil buffer submodule 145c can be used interchangeably.

Further, in at least one implementation, the rendering module 145b can then apply a distortion algorithm to the area of the scene that was previously loaded into the stencil buffer. In at least one implementation, applying the distortion algorithm to the area that was loaded into the stencil buffer can create the appearance of a translucent surface positioned over the area. The rendering module 145b can then communicate 120b the rendered three-dimensional

model 105 to the user interface module 135, which, in turn, can send the information 120c to the display 100, as discussed more fully below.

For example, Figure 2 depicts a scene 250 from a three-dimensional model 105 that has been rendered by a design software application 140. As shown, scene 250 depicts a vase and pedestal 230 on one side of a wall 240, and a chair 220 on the other side of the wall 240. Figure 2 further shows that wall 240, in turn, is comprised of a translucent panel 200 and a clear glass panel 210, whereby the bottom portion of the chair 215 is clearly visible through the clear glass panel 210, while the top portion of the chair 205 is being distorted by the translucent panel 200. One will understand that the translucent panel 200 can be comprised of any material (or arrangement thereof) that partially obscures light from passing through.

One will appreciate that the level of distortion that the translucent panel 200 creates can vary widely depending upon the characteristics of the translucent panel 200 selected by the user. For example, the translucent panel 200 can comprise a heavily frosted glass pane, such that top of the chair 205 is heavily distorted and barely visible to a user of the design software application 140. In contrast, the translucent panel 200 can comprise an only slightly frosted glass pane such that the top of the chair 205 is only slightly distorted to the user.

Additionally, in at least one implementation the translucent panel 200 can further distort the view of objects on the opposing side of the panel 200 by incorporating a color. For example, the user may have selected a translucent panel 200 may include a green tint (not shown). One will appreciate that the green tint can further influence (beyond basic frosting) the appearance of the top of the chair 205 through the translucent panel 200. Similarly, in at least one implementation, if the color of the translucent panel 200 is pronounced enough, it may be difficult or impossible to distinguish the actual color of the top of the chair 205 through the translucent panel 200, because the distorted image of the top of the chair 205 will appear to be the color of the translucent panel 200.

When rendering a scene 250 within a three-dimensional model 105, in at least one implementation, the rendering module 145b can determine what objects and area on the opposing side of the translucent panel 200 are visible to a user through the translucent panel 200. For example, the rendering module 145b can determine what areas on the opposing side of the translucent panel 200 are visible to the user based upon the position of the user's perspective within the three-dimensional model 105. In at least one implementation, the

rendering module 145b relies upon the stencil buffer submodule 145c to determine what objects and area on the opposing side of a translucent panel 200 are visible to a user. For instance, an exemplary stencil buffer submodule 145c can perform this function by loading the area of the scene 250 encompassed by the translucent panel 200.

5 Along these lines, Figure 3 depicts the translucent panel 200 of Figure 2 loaded into a stencil buffer 300. In the depicted implementation, the stencil buffer 300 represents the area that the translucent panel 200 comprises as being black. One will understand that conventional stencil buffers generally use integer values to represent the contents of the buffer. For the sake of clarity, however, Figure 3 depicts the loaded content as simply being
10 black.

 In addition, Figure 3 shows that the stencil buffer 300 does not include the area comprising the vase and pedestal 310, since they are in front of the panel 200. That is, objects that are in front of the translucent panel 200 (with respect to the user's perspective) will not be loaded into the stencil buffer 300. One will appreciate that objects (e.g., the vase and stand)
15 in front of the panel 200 are not distorted by the panel 200; similarly, the portion of the panel 200 behind the object(s) need not be loaded into the stencil buffer 300 since such objects in front of the panel 200 obscure that portion of the panel 200 to the user.

 In any event, after loading the visible area encompassed by translucent panel 200 into the stencil buffer, the rendering module 145b can then render the portions of the scene that are
20 not loaded into the stencil buffer 300. For example, Figure 4 depicts a rendering of scene 250 showing the vase and stand, the unobscured portion of the chair, etc. As previously mentioned, these objects (or portions thereof) are rendered because they were not loaded into the stencil buffer. After rendering these viewable objects, the inventive software then proceeds to render those items obscured by the translucent panel 200.

25 In at least one implementation, the rendering module 145b does not immediately transmit the scene depicted in Figure 4 to the user interface module 135 to then be sent to the display 100. Rather, in at least one implementation, the inventive software uses the scene depicted in Figure 4 as an intermediary step in rendering a complete scene (as depicted in Figure 2). For example, prior to sending a scene 250 to be depicted on the display 100, the
30 rendering module 145b can apply a distortion effect to the translucent panel 200.

In at least one implementation, the inventive software first applies a mask 500 (e.g., via a mask submodule 145d) to the areas of the scene 250 that are not being distorted. For example, Figure 5A depicts the area that the translucent surface comprises and a mask 500 that has been applied to the rest of the scene 250 (i.e., the outside surfaces of the vase and pedestal, and all other aspects of the scene outside the translucent panel 200). One will appreciate that Figure 5A illustrates the top of the chair 205 without distortion because the distortion algorithm has not yet been applied.

Upon masking, the rendering module 145b can then apply the distortion algorithm as appropriate to the translucent panel 200 by blurring the area that the translucent panel 200 comprises. In at least one implementation, this can be accomplished by the rendering module 145b applying a blurring algorithm, such that the blurring algorithm samples each pixel within the panel 200 and causes each pixel to blend the contents of a certain number of neighboring pixels into itself. One will understand that causing each pixel to blend with a certain number of neighboring pixels can cause the blended pixels to appear blurry and distorted.

Furthermore, in at least one implementation this blending effect can be implemented by sampling the colors of a certain number of pixels surrounding a target pixel, performing a blending of the color values, and applying the blended color to the target pixel. One will understand that there are a number of methods that can be used to blend the color values in a blurring effecting including, but not limited to, Gaussian blurs, linear blurs, weighted average blurs, and other known blur functions.

In at least one implementation, the inventive software can apply the blurring effect by randomly sampling a number of pixels surrounding a target pixel, and using the contents of the randomly sampled pixels to create a blurring effect. For example, in contrast to sampling every pixel within a five-pixel radius of the target pixel, the rendering module 145b can randomly sample a select number of pixels within the radius, and use those pixels to create the blurring effect. One will understand that performing the blurring effect by randomly sampling a smaller number of pixels instead of sampling every pixel within a specific radius can cause the blurring effect to require less system resources.

Further, in at least one implementation, the rendering module 145b can be adapted to ignore pixels that are covered by the mask 500 when creating a distortion effect. For

example, when applying a blurring effect to a target pixel within the translucent panel 200, the rendering module 145b can be adapted to ignore any sampled pixels that are covered by the mask 500. For instance, in at least one implementation, the rendering module 145b can determine that a sampled pixel is covered by the mask 500 if the sampled pixel is the same color as the mask 500.

One will appreciate that using the mask 500 to prevent the sampling of pixels that are outside of the translucent panel 200 can aid in limiting the number of artifacts that appear within the translucent panel. For instance, without the mask 500, the blurring effect can blend into a target pixel color information from sampled pixels that are not part of the translucent panel 200. In some cases, incorporating color information from pixels that are not part of the translucent panel 200 into a target pixel can cause artifacts to appear – such as causing the vase and pedestal 230 to have a blurry or distorted perimeter.

For example, if the rendering module 145b were to perform a blurring effect on a frosted glass window that is set within a wood frame, and if the wood frame was not covered by a mask, pixels representing the wood frame can be blended into pixels that are on the edge of the frosted glass window. One will understand that blending pixels that represent wood into pixels that represent frosted glass can lead to the creation of artifacts within the translucent surface. In at least one implementation, covering the areas of the scene 250 that are not part of the translucent panel 200 with a mask 500 can prevent the appearance of artifacts within the translucent panel 200 by preventing the rendering module 145b from sampling pixels that are covered by the mask 500.

Figure 5B depicts the translucent panel 200 after the rendering module 145b has applied the distortion effect. In particular, Figure 5B shows the same scene as illustrated in Figure 5A, except that the top of the chair is now blurred. Specifically, Figure 5B shows that the rendering module 145b did not apply the distortion algorithm to the vase and pedestal since they are protected by the mask 500. In contrast, the rendering module 145b did apply the distortion algorithm to the top of the chair 205, and, as a result, the chair appears to have been distorted by the translucent panel 200.

In general, one will appreciate that the entire area encompassed by translucent panel 200 (excluding the areas covered by the mask 500), will appear to be distorted. In other words, the distortion algorithm will not only distort objects that are on the opposing side of

the translucent panel 200, such as the top of the chair 205. The distortion algorithm will also distort the translucent panel 200 itself, such that the entire panel appears to have a distortion effect associated with it. In at least one implementation, the rendering module 145b can distort the entire translucent panel 200 by applying a blurring effect to the entire panel.

5 Additionally, in at least one implementation, the translucent panel 200 may comprise a pattern – for example, a logo that has been etched into a frosted glass panel. In at least one implementation, the blurring effect can blur the translucent panel 200 without distorting the pattern. For example, in at least one implementation, a mask can be applied to the pattern to prevent it from being blurred. Additionally, in at least one implementation, the translucent
10 panel 200 can be blurred without the pattern and then the pattern can be added to the translucent panel 200 using an alpha channel.

Furthermore, the distortion panel may comprise a tint or color. In at least one implementation, if the translucent panel 200 of Figure 5B comprises a certain tint, the rendering module 145b can distort the panel 200, including the top of the chair 205, by
15 biasing the blur algorithm to include the color or tint. In at least one implementation, the rendering module 145b can bias the blurring algorithm by detecting the color or tint of the translucent panel 200 and adding that color or tint to every pixel that is blurred.

Additionally, in at least one implementation, the rendering module 145b can determine the material of which the translucent panel 200 is comprised. One will understand that
20 different materials will influence the level of translucence and distortion that a translucent panel 200 causes. Further, the rendering module 145b can apply distortion effects that are unique to different materials. For example, the rendering module 145b can apply a distortion algorithm that only lightly distorts the translucent panel 200 if the panel 200 is lightly frosted glass. In contrast, if the translucent panel is heavily frosted, blue tinted glass, then the
25 rendering module 145b can apply a distortion algorithm that heavily distorts the translucent panel 200 and incorporates the proper amount of blue tinting into the distortion. One will understand that the rendering module 145b can incorporate similar adjustments to the distortion algorithm to reflect a variety of translucent materials.

One will also appreciate that the distorted translucent panel 200 of Figure 5B can be
30 combined with the rendered scene 250 from Figure 4 to generate the completed scene 250 of Figure 2. In such a case, the rendering module 145b can then send the information 120b to

the user interface module 135, which can in turn communicate the information 120c to the display.

Accordingly, Figures 1-5B and the corresponding text illustrate or otherwise describe one or more components, modules, and/or mechanisms for efficiently and realistically dealing with objects behind a translucent surface 200 within a three-dimensional model 105. One will appreciate that implementations of the present invention can also be described in terms of flowcharts comprising one or more acts for accomplishing a particular result. For example, Figures 6 and 7 and the corresponding text describe acts in a method for creating translucent surfaces 200 within a three-dimensional model. The acts of Figures 6 and 7 are described below with respect to the components and modules of Figures 1-5B.

For example, Figure 6 illustrates that a method in accordance with an implementation of the present invention for creating translucent surfaces within a three-dimensional model can comprise an act 600 of rendering a three-dimensional model. Act 600 can include rendering a three-dimensional model of a layout of a design space within an architectural design environment. For example, Figure 2 shows a scene 250 from a three-dimensional model 105 that has been rendered.

Additionally, Figure 6 shows that the method can comprise an act 610 of identifying a portion of the three-dimensional model 105. Act 610 can include identifying a first portion of the three-dimensional model that is positioned behind a translucent surface within the three-dimensional model. For example, Figure 5A shows an identified portion of the three-dimensional model – a top of a chair 205 – that is at least partially viewable through the translucent surface from the perspective of the user.

Furthermore, Figure 6 shows that the method can comprise an act 620 of rendering a visual distortion effect. Act 620 can include rendering a visual distortion effect for the identified first portion of the three-dimensional model. For example, Figure 2 shows a visual distortion that has been rendered for a portion 200 of the three-dimensional model 105.

Further still, Figure 6 shows that the method can comprise an act 630 of displaying the three-dimensional model. Act 630 can include displaying on a display device the three-dimensional model comprising the translucent surface, wherein the three-dimensional model is displayed behind the translucent surface in a distorted form based on the rendered distortion

effect. For example, Figure 1 shows a computer display 100 displaying a three-dimensional model 105 that comprises a translucent surface.

Figure 7 shows that an additional or alternative method in accordance with an implementation of the present invention for creating visual distortions through translucent structures can comprise an act 700 of rendering a three-dimensional model. Act 700 can include rendering a three-dimensional model of a layout of a design space within an architectural design environment. For example, Figure 2 shows a scene 250 from a three-dimensional model 105 that has been rendered.

In addition, Figure 7 shows that the method can comprise an act 710 of identifying a translucent surface. Act 710 can include identifying a translucent surface within the three-dimensional model. For example, Figure 5A shows a portion of the three-dimensional model – a view of a top of a chair 205 – has been identified as being a translucent surface.

Furthermore, Figure 7 shows that the method can comprise an act 720 of performing a visual distortion effect. Act 720 includes performing a distortion effect on the translucent surface, wherein performing the distortion effect comprises performing acts 730 and 740. For example, Figures 2 and 5B shows a distortion effect applied to the top of a chair 205.

Additionally, Figure 7 shows that the method can comprise an act 730 of masking a portion of the three-dimensional model 105. Act 730 can include masking a portion of the three-dimensional model from the translucent surface. For example, Figure 5A shows a portion 500 of the three-dimensional model 105 masked from the translucent surface.

Further, Figure 7 shows that the method can comprise an act 740 of blurring one or more pixels. Act 740 can include blurring one or more of the pixels of the translucent surface. For example, Figure 5B shows the translucent surface 200 with one or more blurred pixels.

Further still, Figure 7 shows that the method can comprise act 750 of displaying the three-dimensional model. Act 750 can include displaying to a user the three-dimensional model, including the translucent surface. For example, Figure 1 shows a computer display 100 displaying a three-dimensional model 105 that comprises a translucent surface.

Accordingly, Figures 1-7 provide a number of components, schematics, and mechanisms for creating translucent surfaces within a design software application 140. One will appreciate that the concepts described herein can provide a number of advantages related to the selection of materials and the layout of buildings that have been modeled using the

present invention. For example, at least one implementation of the present invention allows a user a much greater sense of realism when attempting to visually determine the effect that a variety of translucent materials can have on a particular room. The benefits provided by the software described herein enable a designer to, for example, simply and efficiently select a proper amount of frosting to apply to a glass surface that is being inset into the door of a doctor's examination room. In this case, the frosted glass surface should provide sufficient privacy but still allow light to enter the room through the surface. The inventive software components described herein would enable the designer to accurately select the correct level of frosting without unnecessary delays and re-ordering of materials.

The embodiments of the present invention may comprise a special purpose or general-purpose computer including various computer hardware components, as discussed in greater detail below. Embodiments within the scope of the present invention also include computer-readable media for carrying or having computer-executable instructions or data structures stored thereon. Such computer-readable media can be any available media that can be accessed by a general purpose or special purpose computer.

By way of example, and not limitation, such computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to carry or store desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer. When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or a combination of hardwired or wireless) to a computer, the computer properly views the connection as a computer-readable medium. Thus, any such connection is properly termed a computer-readable medium. Combinations of the above should also be included within the scope of computer-readable media.

Computer-executable instructions comprise, for example, instructions and data which cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited

to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in
5 all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

10

15

20

25

30

CLAIMS

I claim:

1. In a computerized architectural design environment in which a design program is loaded into memory and processed at a central processing unit, a computer-implemented method for creating translucent surfaces within a three-dimensional model, the method comprising:
 - rendering a three-dimensional model of a layout of a design space within an architectural design environment;
 - 10 identifying a first portion of the three-dimensional model that is positioned behind a translucent surface within the three-dimensional model;
 - rendering a visual distortion effect for the identified first portion of the three-dimensional model; and
 - 15 displaying on a display device the three-dimensional model comprising the translucent surface, wherein the three-dimensional model is displayed behind the translucent surface in a distorted form based on the rendered distortion effect.
2. The method as recited in claim 1, wherein rendering a visual distortion effect for the identified first portion of the three-dimensional model comprises:
 - 20 identifying a target pixel that is to be distorted; and
 - selecting one or more other pixels to blur into the target pixel.
3. The method as recited in claim 2, further comprising:
 - 25 masking a second portion of the three-dimensional model that surrounds the translucent surface.
4. The method as recited in claim 3, further comprising:
 - determining whether one of the selected one or more other pixels originated from the masked second portion of the three-dimensional model; and
 - 30 excluding the one of the selected one or more other pixels from being blurred into the target pixel.

5. The method as recited in claim 3, wherein masking the second portion of the three-dimensional model comprises:

assigning a specific attribute to the second portion of the three-dimensional
5 model.

6. The method as recited in claim 5, wherein assigning a specific attribute to the second portion of the three-dimensional model comprises rendering the second portion in a specific uniform color.

7. The method as recited in claim 2, further comprising:

identifying an attribute associated with the translucent surface; and
adjusting the visual distortion effect based upon the attribute.

8. The method as recited in claim 7, further comprising:

identifying a color that is associated with the translucent surface; and
biasing the visual distortion effect to incorporate the color.

9. The method as recited in claim 7, further comprising:

identifying a magnitude of translucence that is associated with the translucent
surface; and
biasing the visual distortion effect to incorporate the magnitude of
translucence.

10. The method as recited in claim 1, further comprising:

loading the first portion of the three-dimensional model into a stencil buffer,
wherein the first portion is not initially rendered.

11. In a computerized architectural design environment in which a design program is

loaded into memory and processed at a central processing unit, a computer-implemented

method for creating visual distortions effects through translucent structures, the method comprising:

rendering a three-dimensional model of a layout of a design space within an architectural design environment;

5 identifying a translucent surface within the three-dimensional model;

performing a distortion effect on the translucent surface, wherein performing the distortion effect comprises:

masking a portion of the three-dimensional model from the translucent surface; and

10 blurring one or more of the pixels of the translucent surface; and

displaying to a user the three-dimensional model, including the translucent surface.

12. The method as recited in claim 11, wherein blurring one or more of the pixels of the translucent surface comprises:

identifying a target pixel that is to be distorted; and

selecting one or more other pixels to blur into the target pixel.

13. The method as recited in claim 12, further comprising:

20 determining whether one of the selected one or more other pixels originated from the masked portion of the three-dimensional model; and

excluding the one of the selected one or more other pixels from being blurred into the target pixel.

25 14. The method as recited in claim 13, wherein masking the portion of the three-dimensional model from the translucent surface comprises:

assigning a specific attribute to the masked portion of the three-dimensional model.

30

15. The method as recited in claim 14, wherein assigning a specific attribute to the masked portion of the three-dimensional model comprises rendering the masked portion in a specific uniform color.

5 16. The method as recited in claim 11, further comprising:
identifying an attribute associated with the translucent surface; and
adjusting the distortion effect based upon the attribute.

17. The method as recited in claim 16, further comprising:
10 identifying a color that is associated with the translucent surface; and
biasing the distortion effect to incorporate the color.

18. The method as recited in claim 16, further comprising:
identifying a magnitude of translucence that is associated with the translucent
15 surface; and
biasing the distortion effect to incorporate the magnitude of translucence.

19. The method as recited in claim 11, further comprising:
loading the translucent surface into a stencil buffer, wherein the translucent
20 surface is not initially rendered.

20. A computer program product for use at a computer system, the computer program product for implementing a method for creating translucent surfaces within a three-dimensional model, the computer program product comprising one or more computer storage
25 media having stored thereon computer-executable instructions that, when executed at a processor, cause the computer system to perform the method, including the following:
rendering a three-dimensional model of a layout of a design space within an architectural design environment;
identifying a first portion of the three-dimensional model that is positioned
30 behind a translucent surface within the three-dimensional model;

rendering a visual distortion effect for the identified first portion of the three-dimensional model; and

displaying on a display device the three-dimensional model comprising the translucent surface, wherein the three-dimensional model is displayed behind the translucent surface in a distorted form based on the rendered distortion effect.

5

10

15

20

25

30