

METHOD AND SYSTEM FOR EFFICIENT MODELING OF SPECULAR REFLECTION

BACKGROUND OF THE INVENTION

5 The Field of the Invention

The present invention relates generally to computer-generated graphics, and more specifically relates to modeling and rendering specular reflection of light.

Background and Relevant Art

As computerized systems have increased in popularity so have the range of
10 applications that incorporate computational technology. Computational technology now extends across a broad range of applications, including a wide range of productivity and entertainment software. Indeed, computational technology and related software can now be found in a wide range of generic applications that are suited for many environments, as well as fairly industry-specific software.

15 One such industry that has employed specific types of software and other computational technology increasingly over the past few years is that related to 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
20 commercial space. For example, a designer might use 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.

While three-dimensional rendering is becoming a more common feature in
25 CAD programs, three-dimensional rendering is a fairly resource intensive process. For example, a traditional rendering program can take anywhere from several minutes to several hours to appropriately render all of the lighting and shading effects of a given space with accuracy. This may be particularly inconvenient to a designer who has to wait for the scene to render after making a change to the layout of the scene.
30 Alternatively, some rendering programs may use methods of rendering that result in less realistic images to speed up the rendering and use fewer resources. Such programs may do so by, for example, rendering fewer features within the scene or by using pre-rendered elements that do not necessarily correspond with the actual scene being rendered.

One such lighting effect that can require intensive resources to properly incorporate into a three-dimensional model is specular reflections. Specular reflection of light occurs when light hits a surface that reflects the light in a relatively narrow range of directions, forming the appearance of shiny spots on an object. Proper rendering of specular reflection can contribute significantly to the realism of a three-dimensional model.

Many conventional methods of calculating specular reflection have various disadvantages. For instance, when multiple light sources are present in a scene, many conventional methods calculate specular reflection for each individual light and combine the effect of the individual lights. For multiple lights, the computation load can increase linearly with additional lights. Because the number of lights is typically large in certain settings, such as in offices and department stores, rendering can slow down (many-fold) in these applications.

A second limitation with conventional methods lies in the algorithm used to calculate specular reflection. Some conventional methods model the specular reflection value for each individual light based on the equation $S_i = I_i k_s (R_i \cdot V_i)^n$. In this equation, I_i is the intensity of light source i , k_s is a specular reflection constant, and $R \cdot V$ is the dot product of reflection vector R and viewing vector V shown in FIG. 1. The parameter n is a variable reflecting the smoothness of a surface. Because the incidence angle α between the incident light vector L and the normal vector N equals the reflection angle α between R and N , R can be computed as $R = 2(L \cdot N) N - L$. These conventional methods require calculation of not only the viewing vector V , but also the reflection vector R , which often changes at different points of a surface. The calculation of the reflection vector R adds to the computational burden for the simulation of specular reflection.

An additional problem exists in conventional methods using the foregoing algorithm. These conventional methods can require calculating the dot product of reflection ray vector R and viewing vector V , or the cosine value of the viewing angle, $\cos(\beta)$. These methods then can typically raise the value to the power of 200 or higher to simulate the appearance of shiny surfaces. These methods need to repeat the same calculation for each point on an object's surface, and for every viewing perspective in a dynamic viewing situation. These methods can substantially increase the computational load and slow down simulation of the object in dynamic scenes.

The foregoing problems make many of the conventional methods too slow to be practical in real-time rendering of dynamic scenes, such as in a virtual walk-through of a three-dimensional architectural model, video games, or other virtual environments. Computer generated graphics for these settings typically require
5 rendering rate of 30 frames per second or higher to achieve smooth motion and realistic appearance.

One method intended to circumvent the foregoing problems is the “baking” method. The baking method involves pre-calculating the specular reflection of an object in a particular setting, usually with a fixed lighting condition. The convention
10 software then applies the same reflection of the object to different frames of a video for a dynamic scene, regardless of the changes of the incident vector L , reflection vector R , or the viewing vector V . Although this method is computationally simpler than conventional methods that calculate specular reflection for each frame, the realism of a rendered object is substantially inferior when object placement and the
15 viewing angle deviate from the original setting.

Some conventional methods used in video games utilize “ambient occlusion” techniques to improve realism. The results, however, are not realistic due to the use of a single, global ambient light source, and conventional design spaces tend to incorporate a plurality of light sources. In addition, ambient occlusion techniques
20 typically do not allow for dynamic differential of lighting in different areas of the layout or scene, due to the single, global light source nature. Ambient occlusion techniques usually have no ability (or are significantly limited) to turn off lighting in one area of the scene, and turn on lighting in another area.

Accordingly, there are several disadvantages in the art that can be addressed.
25

BRIEF SUMMARY OF THE INVENTION

Implementations of the present invention overcome one or more of the foregoing or other problems in the art with systems, methods, and apparatus configured to allow efficient rendering of realistic specular reflection of objects in a
30 design software application. In particular, one or more implementations of the present invention allow for simulation of specular reflection without having to perform a separate calculation for each light in an environment with multiple lights. In addition, one or more implementations of the present invention have the ability to model specular reflection with sufficient realism while turning off lighting in one area of the

scene, and while turning on lighting in another area. In general, one or more implementations of the invention provide fast and realistic simulation of specular reflection, allowing real-time rendering of videos and dynamic scenes in applications such as virtual walk-throughs of interior design spaces, video games, and other virtual environments.

For example, a method in accordance with one or more implementations of rendering a specular effect within a three-dimensional model can include creating a map that comprises a representation of at least one light source within a three-dimensional model. The method can also include casting a viewpoint ray to an object surface point. In such a case, the viewpoint ray comprises a ray extending from a user perspective within the three-dimensional model. In addition, the method can include casting a reflection ray of the viewpoint ray. Furthermore, the method can include identifying an intersection point between the reflection ray and the map. Still further, the method can include calculating the specular reflection of the object surface point based on the intersection point.

In an additional or alternative implementation, a method for rendering a specular effect within a three-dimensional model can include rendering a three-dimensional model. In such a case, the three-dimensional model comprises at least one light source. In addition, the method can include creating at least one map of a plane within the three-dimensional model. As such, the at least one map comprises at least one representation of the at least one light source. Additionally, the method can include casting a ray from a surface within the three-dimensional model to the at least one map. Furthermore, the method can include identifying an intersection point between the ray and the at least one map. Still further, the method can include calculating the specular reflection of the surface based on the intersection point.

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
5 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 a schematic diagram showing the variables involved in modeling specular reflection in the prior art;

Figure 2 depicts an architectural schematic diagram of a computer system for
10 rendering specular effect within a three-dimensional model;

Figure 3 illustrates a model of a room including furniture without specular reflection;

Figure 4 illustrates a specular intensity map based on the room illustrated in Figure ;

15 Figure 5 illustrates another specular intensity map similar to that in Figure 4;

Figure 6 illustrates the room of Figure 3, comprising vectors used in the calculation of specular effect;

Figure 7 illustrates the room of Figure 3, comprising another set of vectors used in the calculation of specular effect;

20 Figure 8 illustrates the furniture in the same interior environment as Figure 3, albeit with the furniture now comprising a specular effect ;

Figure 9 illustrates a flowchart of a series of acts in a method in accordance with an implementation of the present invention for rendering specular reflection in computer-generated graphics; and

25 Figure 10 illustrates another flowchart of a series of acts in a method in accordance with an implementation of the present invention for rendering specular reflection in computer-generated graphics.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

30 Implementations of the present invention extend to systems, methods, and apparatus configured to allow efficient rendering of realistic specular reflection of objects in a design software application. In particular, one or more implementations of the present invention allow for simulation of specular reflection without having to perform a separate calculation for each light in an environment with multiple lights.

In addition, one or more implementations of the present invention have the ability to model specular reflection with sufficient realism while turning off lighting in one area of the scene, and while turning on lighting in another area. In general, one or more implementations of the invention provide fast and realistic simulation of specular reflection, allowing real-time rendering of videos and dynamic scenes in applications such as virtual walk-throughs of interior design spaces, video games, and other virtual environments.

For example, one or more implementations of the present invention provide a highly efficient method for rendering specular reflection in an environment having multiple light sources. In particular, one or more implementations use at least one pre-calculated specular intensity gradient to model specular reflection. Additionally, in one or more implementations, a single pre-calculated lighting map can simultaneously account for the effects of multiple light sources. As used within this application, a lighting map is a map that comprises characteristics of at least one light source and is associated with a three-dimensional model such that at least one vector drawn within the three-dimensional model can intersect with the map. This capability can eliminate the need to separately calculate a specular reflection value for each individual light.

Additionally, one or more implementations of the present invention can pre-calculate a lighting map based upon a variety of dynamic lighting conditions. For example, a design software application can pre-calculate a lighting map to reflect which lights within a particular room have been turned on. Further, a design software application can recalculate a lighting map when various lights with the particular room have been turned on or off. By adjusting the effects of individual lights on the lighting map, one or more implementations allow for simulation of dynamic differential lighting in different areas of a layout or scene.

In addition, one or more implementations of the present invention can model specular reflection to simulate a wide range of surface types. For instance, various materials can produce different specular reflection based upon the reflective characteristics of the material. For example, one or more implementations can pre-calculate lighting maps that incorporate a variety of reflective characteristics of surfaces using multiple specular intensity gradients having different rates of attenuation.

This can be computationally efficient because the method can pre-calculate multiple lighting maps for a particular lighting environment. One or more implementations can then use the same lighting maps to model specular refraction for different objects in the same lighting environment. Similarly, one or more
5 implementations can also use the same lighting maps to model specular reflection for varying object positions and viewing angles.

Accordingly, one will appreciate in view of the specification and claims herein that at least one implementation of the present invention provides the ability to render a specular effect within a three-dimensional model. Specifically, at least one
10 implementation of the present invention pre-calculates at least one lighting map that comprises characteristics of the lighting within at least a portion of the three-dimensional model. The design software application can then rely upon the information within the at least one lighting map to render a specular effect in real-time.

Referring now to the figures, Figure 2 depicts an architectural schematic diagram of a computer system for rendering specular effect within a three-dimensional model. In particular, Figure 2 shows user input devices 110 that communicate with the computer system 120, which in turn communicates with a display 100. Figure 2 shows that the user input devices 110 can include any number
15 of input devices or mechanisms, including but not limited to a mouse 112a, a keyboard 112b, or other forms of user input devices (including remote devices, touch screens, etc.,).

In addition, Figure 2 shows that computer system 120 comprises a design software application 140 executed by a processing unit 130. One will appreciate that
25 the processing unit 130 can comprise a central processing unit, a graphics processor, a physics processor, or any other type of processing unit. Figure 2 further shows that the computer system 120 can comprise a storage device 160. In one implementation, storage device 160 can contain, among other things, templates and objects that can be placed within a three-dimensional model 105. These components, in conjunction with
30 processing unit 130, store and execute the instructions of design software application 140.

Figure 2 shows that a user in this case uses the input device(s) 110 to send one or more requests 132a to the computer system 120. In one implementation, the processing unit 130 implements and/or executes the requests from user input devices

110, and application 140 instructions. For example, a user can provide one or more inputs 132a relating to the design and rendering of a three-dimensional model 105 within a design software application 140, as executed by processing unit 130. Figure 2 further shows that the design software application 140 can then pass the inputs 132b
5 to the appropriate modules within the design software application 140.

Ultimately, design application 140 can then send corresponding rendering instructions 132c through the user interface module 142a to display 100. As shown in Figure 2, for example, display 100 displays a graphical user interface in which the user is able to interact (e.g., using the user input devices 110). In particular, Figure 2
10 shows that the graphical user interface can include a depiction of a three-dimensional model 105 of a design space comprising in this case a chair, a desk, and one or more lighting elements.

One will appreciate in view of the specification and claims herein that the user interface module 142a can provide to the user an option to make design changes to the three-dimensional model 105. In one implementation, for example, upon receiving a
15 request for some modification, the user interface module 142a can communicate the request to the design module 142b.

One will appreciate that the design module 142b 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, adjust light sources within the three-dimensional model 105, or change parameters relating to the three-dimensional model 105. In some cases, this can include the design module 140b communicating with the storage device 160 to retrieve, among other things, templates and objects that can be placed
25 within a three-dimensional model 105.

After receiving and processing a user input/request, the design module 142b can then send a corresponding request to the rendering module 142c for further processing. In one implementation, this further processing includes rendering module 142c rendering the depiction of the three-dimensional model 105 shown in this case on the display 100 in Figure 2. One will appreciate that the rendered depiction can include shading and lighting effects within the three-dimensional model 105. The rendering module 142c can then communicate with the storage device 160. In one implementation, information that is accessed frequently and is speed-sensitive will be stored within a high-speed memory.
30

Figure 2 further shows that the rendering module 142c can communicate with a specular reflection module 142d. In one implementation, the specular reflection module 142d can calculate the specular effects, including specular reflections, for a three-dimensional model 105. Specifically, the specular reflection module 142d can calculate the effect of multiple light sources on specular effects within the three-dimensional model 105 and store those calculations within at least one lighting map. The rendering module 142c, using the calculations, can then render specular effects within the three-dimensional model 105 in real-time without respect to the number of light sources that are contained within the three-dimensional model 105.

When a user makes changes to the lighting sources within the three-dimensional model 105, the specular reflection module 142d can automatically recalculate the at least one lighting map, taking into account the changes to the light sources. In at least one implementation, the specular reflection module 142d can calculate the specular effects within either the entire three-dimensional model 105 or just a portion of the three-dimensional model 105.

Once the specular reflection module 142d has initially calculated at least one lighting map, in at least one implementation, the at least one lighting map will not be recalculated until a design change is made to the three-dimensional model 105. Additionally, in at least one implementation, the calculation time for creating a lighting map may increase with the number of light sources present within the three-dimensional model. Once the lighting map is calculated, however, the rendering module 142c can render a three-dimensional model 105, including specular effects, in real-time.

Figure 3 depicts a three-dimensional model 105 of a room 300 containing a desk 310, a chair 320, and several lights 301, 302, 303. In this case, Figure 3 does not depict any specular effects within the three-dimensional model 105. In at least one implementation, the room 300 of Figure 3 can be part of a larger three-dimensional model 105 of a building.

Additionally, in one or more implementations of the present invention, the specular reflection module 142d can create a map that contains information relating to each of the lighting sources 301, 302, 303. Further, the information relating to each of the light sources can comprise gradients of specular intensity and locations of light sources with respect to a plane in the three-dimensional model 105.

For example, Figure 4 shows a glossy lighting map 450 including indications 401, 402, 403 of the light sources 301, 302, 303 respectively from Figure 3. In at least one implementation, the indications 401, 402, 403 of the light sources 301, 302, 303 comprise gradients representing the diffusion of each respective light source over a glossy surface. Further, in at least one implementation, the gradients are brightest in the center and gradually grow dimmer moving radially outward from the center point of the gradient.

Additionally, in at least one implementation, the specular reflection module 142d creates the glossy specular map 450 based upon a plane 410 within the three-dimensional model 105 of Figure 3. In at least one implementation, the specular reflection module creates the glossy specular map 450 based upon a plane 410 within the three-dimensional model 105 of Figure 3 that contains the actual light sources 301, 302, 303.

In contrast, in at least one implementation, the specular reflection module 142d can select a plane that does not intersect any light sources. For example, in at least one implementation, the light sources may not be on a single plane. The specular reflection module 142d can select a plane 410 substantially parallel to the floor or ceiling of the environment and create a glossy lighting map 450 that models all of the light sources 301, 302, 303 within the model 105, even if those light sources do not intersect the plane.

One will understand that in at least one implementation, the specular reflection module 142d is not limited to selecting planes with specific orientations. For example, the specular reflection module 142d can select a plane that is not parallel to the floor or ceiling and still create a lighting map 450 that models at least one light source.

When the light sources are not on a single plane, in at least one implementation, the specular reflection module 142d can create a map of the characteristics of the light by calculating the normal projections of the lights on the plane. In at least one implementation of the invention, the specular reflection module 142d maps the characteristics of the light sources 301, 302, 303 onto the plane 410 by other methods, such as extending from a light source a line that is normal to the floor or ceiling of a room, and identifying the intersection of the line on the plane as the location of the light source.

Additionally, in at least one implementation, the specular reflection module 142d can use two or more planes for mapping light locations. This can be particularly useful when lights are situated on two or more elevations within a room 300, such as in a room 300 with a number of ceiling lights inset into the ceiling and a number of pendant lights hanging from the ceiling.

For example, the specular reflection module 142d can create two different planes that are substantially parallel to the floor and/or ceiling of the interior environment. Further, in at least one implementation, the specular reflection module 142d can place one plane at the ceiling, such that it intersects the ceiling lights, and place the other ceiling such that it intersects the pendent lights. Further still when additional lights are situated off the two planes, in at least one implementation, the specular reflection module 142d can map the additional lights onto the closest plane by the lights' normal projections on the plane.

Figure 6 shows that in room 300 at least one implementation of the invention identifies a plane 410 at light sources 301, 302, and 303. In this example, plane 410 is substantially parallel to the floor and ceiling of the room 300. Light sources 301, 302, and 303 are mapped to points 422, 424, and 426, respectively as shown in Figure 4.

After identifying a plane and the locations of the light sources on the plane, in at least one implementation, the specular reflection module 142d can calculate an attenuating gradient of specular intensity 401, 402, 403 on the map 450. In at least one implementation, the specular intensity values of the attenuating gradient can provide a basis for simulating the specular reflection on objects to be imaged in the environment.

In at least one implementation, specular intensity attenuates as the distance from the mapped location of a light increases. For example, as depicted in Figure 4, light source 301 has been mapped at point 422. In at least one implementation, the specular intensity 401 associated with the light 301 decreases radially. For instance, in at least one implementation, the specular reflection module 142d can model the attenuating gradient of specular intensity as a 3-D Gaussian function, where the origin of the (x,y) plane coincides with the mapped location of the light source.

Additionally, in at least one implementation, the specular reflection module can use functions other than the 3-D Gaussian function to model a specular intensity gradient. For instance, at least one implementation can use a 3-D cosine function, where the peak of the function coincides with the location the light source on the map.

Similarly, at least one implementation can empirically modify the shape of the 3-D cosine function to achieve a specular intensity gradient effecting realistic specular reflection. For instance, at least one implementation can raise the cosine function to the power of n to simulate the effect of surfaces of different reflective properties.

5 Additionally, in at least one implementation, the specular intensity at a point on the map depends on the distance between the location of the light within the three-dimensional model 105 and the plane that the specular reflection module 142d selected. For example, in Figure 6 the specular reflection module 142d selected a plane 410 that intersected with the lights 301, 302, 303, such that the specular
10 intensity of map 450 is at a relative maximum. In contrast, if the specular reflection module 142d selected a plane 410 that was medial to the lights 301, 302, 303 and the floor of the room 300, then the specular intensity of the map would be appropriately diminished.

 Similarly, in at least one implementation the specular intensity of the
15 attenuating gradient 401, 402, 403 may depend on the brightness of the light source 301, 302, 303, such that a brighter light source leads to higher specular intensity values. In these implementations where specular intensity depends on the brightness of the light source or its distance from the plane, a brightness or distance factor may be added to or multiplied by the specular intensity values.

20 Furthermore, as appreciated by one skilled in the art, the specular intensity value can be a scalar for rendering of a gray scale image. In color image rendering, the specular intensity value can comprise a vector. For instance, it can be a vector [R, G, B] to represent the red, green, blue light intensity in the RGB color space for additive color rendering. In at least one implementation of the invention, the specular
25 intensity value can be a vector of [C,M,Y,K] representing the colors cyan, magenta, yellow, and key (black) in a subtractive CMYK color model. One will understand that the color can be stored within a variety of models and still be within the scope of the present invention.

 In at least one implementation, one of the advantages of the present invention
30 is that it can efficiently simulate specular reflection of objects in environments with multiple light sources. According to the present invention, a single lighting map 450 accounts for the effects of multiple light sources 301, 302, 303. For instance, in at least one implementation of the present invention, two or more light sources' effects

401, 402 are combined 430 in an additive fashion (e.g., at point 430), generating an intensity map with multiple peaks and/or troughs between peaks.

Additionally, in at least one implementation of the present invention, the rate at which a specular intensity gradient attenuates is determined by the reflectivity or glossiness of the surface of the object to be imaged. In such implementations, a more reflective or glossier surface tends to have a specular intensity gradient attenuating at a higher rate than a duller or rougher surface. Accordingly, at least one implementation generates multiple lighting maps for the calculation of specular reflection for different types of surfaces.

For example, Figure 5 illustrates a matte lighting map 550 similar to the glossy lighting map 450 depicted in Figure 4, but showing specular reflection gradients 501, 502, 503 adapted to calculate the specular reflection of surfaces that are less reflective than those in FIG. 4. In at least one implementation, the rendering module 142 can use this matte lighting map 550 to give a surface the appearance of a matte look.

In at least one implementation of the present invention, the specular reflection module 142d can pre-calculate both the glossy lighting map 450 in FIG. 4 and the matte lighting map 550 in FIG. 5. The two maps 450, 550 can then be repeatedly used in a real-time calculation of specular reflection of shinier surfaces and duller surfaces, respectively. One will understand that, in one or more implementations, more than two lighting maps can be generated.

For example, Figure 6 is a diagram illustrating a method for calculating a specular effect at point 610 on the desk 310 within the three-dimensional model 105. In the implementation depicted by Figure 6, a user's perspective 620 within the three-dimensional model 105 is such that the viewing ray 625 extending from the user's perspective 620 intersects with point 610 on the desk 310.

In at least one implementation, the specular reflection module 142d calculates the specular effect at point 610 by first determining the reflective properties of point 610. As depicted in Figure 6, point 610 is positioned on the surface of desk 310 and as such likely comprises glossy properties. Once the specular reflection module 142d determines the reflective properties of point 610 it can select the proper lighting map. When the reflective properties of a particular object's surface falls between the values described by two lighting maps 450, 550, the rendering module 142c can interpolate between the two lighting maps 450, 550 to calculate a specular effect for the object.

In the example of Figure 6, for the sake of clarity, the specular reflection of the surface of the desk 310 matches the glossy lighting map 450.

The specular reflection module 142d can then determine the specular effect at point 610 by calculating a reflection ray 630 of the viewing ray 625. Various methods
5 are known in the art for calculating a reflection vector. For example, the specular effect module 142d could use $r = v - 2(v \cdot n)n$, where v is the viewing ray and n is the normal vector 640 of point 610.

After calculating the reflection ray 630, the specular reflection module 142d can determine the intersection point 430 of the reflection ray 630 and the glossy
10 lighting map 450. As used within this application, the “intersection point” is the location where the reflection ray 630 and the lighting map 430 intersect. In at least one implementation, the intersection point 430 can be calculated by determining where the reflection ray 630 intersects with the plane 410 on which the lighting map 450 was based. In such a case, the intersection point 430 would be the equivalent
15 point on the lighting map 450.

In at least some implementations, the intersection point (e.g., 430) on the map contains specular reflection information that can be applied to the object. For example, in the depicted implementation, the specular reflection module 142d can create the glossy lighting map 450 due to the reflective properties of point 610 on the
20 desk 310. The rendering module 142c can then use information that is stored at intersection point 430 within the glossy lighting map 450 to render a specular effect at point 610 on the desk 310. In at least one implementation, the information at the intersection point (e.g., 430) can comprise an intensity value that the rendering module 142c can use to determine an intensity of specular effect to apply to a certain
25 point.

Figure 4 and Figure 6 illustrate that reflection ray 630 intersects with glossy lighting map 450 at intersection point 430. The specular reflection module 142d can determine the specular effect at point 610 by using the information stored at point 430 within the glossy lighting map 450. One will understand that because point 430 on
30 the glossy map 450 falls within the specular intensity gradients 401, 402 of both light source 301 and light source 302, the specular effect at point 610 is being influenced by both light sources 301, 302.

Additionally, in at least one implementation, the specular reflection module 142d can also account for the distance between point 610 and intersection point 430

when determining the specular effect at point 610. One will understand that the greater the distance that point 610 is away from a particular light source, the relatively less specular effect that will be caused by the particular light source.

In at least one implementation, the specular reflection module 142d can
5 determine that reflection ray 630 intersects with a surface within the three-dimensional model 105 prior to intersecting with the glossy lighting map 450. For example, the reflection ray might intersect with another piece of furniture. In this case, the specular reflection module 142d can determine that there is no specular effect at point 610 because the specular effect is being blocked by a surface.

10 Figure 7 is a diagram similar to that of Figure 6, except that it demonstrates the modeling of specular reflection at point 710 on the chair 320 instead of point 610 on the desk 310. The same principles and methods described above for modeling the reflection of point 610 on the desk 310 also apply here. Because point 710 of chair 320 has a rougher and duller surface than point 610 on the desk 310, this example
15 implementation uses the matte lighting map 550 of Figure 5. The matte lighting map 550 has a lower rate of attenuation, and thus more realistically reflects the appearance of specular effects on the chair 320 surface.

The examples given so far use two predetermined specular intensity gradients. As explained above, at least one implementation of the invention can create more
20 specular gradients to simulate different levels of shininess and surface texture characteristics. Additionally, in at least one implementation, the specular reflection module 142d can determine the intersection of a reflection ray and multiple lighting maps (to determine the “intersection point”). The specular reflection module 142d can then use the information from the multiple lighting maps to interpolate a proper
25 specular effect for a particular surface.

For example, in at least one implementation, the specular reflection module 142d can create multiple lighting maps, each of which contains lighting characteristics from different light sources within the three-dimensional model 105. The lighting maps can each respectively be based upon light sources that are at different elevations
30 within the three-dimensional model 105. In this implementation, the specular reflection module 142d can determine the intersection points (e.g., 430, 530) of a reflection ray and each of the lighting maps. The resulting information derived from the intersection points can then be additively used to determine the specular effect at a surface caused by all of the light sources at the various elevations.

Figure 8 is an illustration of the desk 310 and chair 320 of Figure 6 and Figure 7, except that Figure 8 comprises specular reflections 810 and 820 on the desk 310 and chair 320 respectively. The specular reflection can add a shiny and hard surface quality to the desk 310, and it can give a matte appearance to the chair 320. One will understand that the ability to utilize specular effects within a three-dimensional model 105 can increase the realistic qualities of the model 105.

Accordingly, Figures 1-8 and the corresponding text illustrate or otherwise describe one or more components, modules, and/or mechanisms for rendering specular effects within a three-dimensional model 105. One will appreciate that implementations of the present invention can allow a designer to create an office building with multiple light sources and have the specular effect from each of those light sources rendered within the three-dimensional model 105. Additionally, one will appreciate that rendering the specular effects created by multiple light sources within a three-dimensional model can aid users in the selection of surfaces within a designed building.

For example, Figure 9 illustrates that a method for rendering specular effects within a three-dimensional model can comprise an act 900 of creating a map. Act 900 includes creating a map that comprises a representation of at least one light source within a three-dimensional model. For example, Figures 4 and 5 show different maps that comprise different representations of each light source within the three-dimensional model.

Figure 9 also shows that the method can comprise act 910 of casting a ray to an object. Act 910 includes casting a viewpoint ray to an object surface point, wherein the viewpoint ray comprises a ray extending from a user perspective within the three-dimensional model. For example, Figures 6 and 7 show rays 625 and 725 being cast from user perspective 620 and 720 and being directed towards a desk 310 and a chair 320, respectively.

Additionally, Figure 9 shows that the method can comprise act 920 of casting a reflection ray. Act 920 includes casting a reflection ray of the viewpoint ray. For example, Figures 6 and 7 show reflection rays 630 and 730 being cast as reflections of rays 625 and 725, respectively.

Furthermore, Figure 9 also shows that the method can comprise act 930 of identifying an intersection point 430, 530. Act 930 includes identifying an intersection point 430, 530 between the reflection ray and the map. Figures 4-7 all

show an intersection point between reflection rays 630, 730 and maps 450, 550, respectively (i.e., intersection points 430, 530).

Still further, Figure 9 shows that the method can comprise act 940 of calculating the specular effect. Act 940 includes calculating a specular effect on the object surface point based on the intersection point 430, 530. For example, Figures 6 and 7, and the resulting Figure 8, depicts a three-dimensional model, comprising vectors that can be used in the calculation of specular effect. Specifically, in Figures 6 and 7 the rays 630 and 730 are depicted intersecting with lighting maps 450 and 550, respectively. The inventive software can then use the intersection points 430 and 530 to calculate a specular effect on the object surface.

In addition to the foregoing, Figure 10 shows that a method of rendering specular effects within a three-dimensional model can comprise an act 1000 of rendering a three-dimensional model 105. Act 1000 includes rendering a three-dimensional model 105, the three-dimensional model comprising at least one light source. For example, Figure 3 shows a rendered three-dimensional model having three light sources.

Figure 10 also shows that the method can comprise an act 1010 of creating a map of a plane. Act 1010 includes creating at least one map of a plane within the three-dimensional model, the at least one map comprising at least one representation of the at least one light source. For example, Figures 4 and 5 show different maps that comprise different representations of each light source within the three-dimensional model 105.

Additionally, Figure 10 shows that the method can comprise an act 1020 of casting a ray from a surface. Act 1020 includes casting a ray from a surface within the three-dimensional model to the at least one map. For example, Figures 6 and 7 show rays 630 and 730 being cast to maps 450 and 550, respectively.

Figure 10 also shows that the method can comprise act 1030 of identifying an intersection point (e.g., intersection points 430 and 530). Act 1030 includes identifying an intersection point between the ray and the at least one map. Figures 4-7 show intersection points 430 and 530 between reflection rays 630, 730 and maps 450, 550, respectively.

Furthermore, Figure 10 shows that the method can comprise an act 1040 of calculating the specular effect. Act 1040 can include calculating a specular effect on the surface based on the intersection point. For example, Figure 6, and the resulting

Figure 8, depict a three-dimensional model, comprising vectors that can be used in the calculation of specular effect. Specifically, in Figure 6 the ray 630 is depicted as intersecting with lighting map 450. The inventive software can then use the intersection point 430 to calculate a specular effect on the object surface.

5 Accordingly, Figures 1-10, and the corresponding text, illustrate or otherwise describe a number of components, schematics, and mechanisms for providing for the rendering of specular effects within a three-dimensional model. One will appreciate in view of the specification and claims herein that the components and mechanisms of the present invention provide the ability to model specular reflection with sufficient
10 realism while turning off lighting in one area of the scene and turning on lighting in another area. In general, one or more implementations of the invention provide fast and realistic simulation of specular reflection, allowing real-time rendering of videos and dynamic scenes in applications such as virtual walk-throughs of interior design spaces, video games, and other virtual environments

15 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
20 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
25 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
30 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
5 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 all respects only as illustrative and not restrictive. The scope of
10 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.

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-
5 implemented method for rendering a specular effect within a three-dimensional model, the method comprising:
 - creating a map that comprises a representation of at least one light source within a three-dimensional model;
 - casting a viewpoint ray to an object surface point, wherein the
10 viewpoint ray comprises a ray extending from a user perspective within the three-dimensional model;
 - casting a reflection ray of the viewpoint ray;
 - identifying an intersection point between the reflection ray and the map; and
 - 15 calculating a specular effect on the object surface point based on the intersection point.
2. The method as in claim 1, wherein the representation of the at least one light source comprises at least one attenuating gradient of specular intensity.
20
3. The method as in claim 2, wherein the at least one attenuating gradient comprises two or more gradients with different rates of attenuation.
4. The method as in claim 2, wherein the specular intensity on the map decreases
25 as a distance between a location of the at least one light source and the representation of the at least one light source on the map increases.
5. The method as in claim 2, wherein the specular intensity of the at least one attenuating gradient is defined by a 3-D Gaussian or 3-D cosine function.
30
6. The method as in claim 2, wherein the specular intensity of the at least one attenuating gradient depends on the brightness of the light source.

7. The method as in claim 1, wherein the at least one light source comprises two or more light sources.

5 8. The method as in claim 1, wherein the specular intensity value is a vector of R, G, B values.

9. The method as in claim 1, wherein the specular intensity value is a vector of C, M, Y, K values.

10

10. 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 rendering a specular effect within a three-dimensional model, the method comprising:

15 rendering a three-dimensional model, the three-dimensional model comprising at least one light source;

creating at least one map of a plane within the three-dimensional model, the at least one map comprising at least one representation of the at least one light source;

20 casting a ray from a surface within the three-dimensional model to the at least one map;

identifying an intersection point between the ray and the at least one map; and

25 calculating a specular effect on the surface based on the intersection point.

11. The method as in claim 10, further comprising:

creating at least one attenuating gradient of specular intensity on the map.

30

12. The method as in claim 11, wherein the specular intensity of the at least one attenuating gradient depends on a distance between a light source and the plane.
13. The method as in claim 11, wherein an effect of two or more light sources on the at least one attenuating gradient is cumulative.
14. The method as in claim 10, wherein the plane is substantially parallel to the floor and/or ceiling in the three-dimensional model.
15. The method as in claim 10, wherein the at least one light source is on the plane, and the at least one representation of the at least one light source on the plane is equivalent to the at least one light source's normal projection on the plane.
16. The method as in claim 10, further comprising creating two or more planes, wherein each plane comprises at least one representation of one or more light sources that intersects with each respective plane.
17. The method as in claim 10, further comprising:
- creating two or more maps of the plane within the three-dimensional model;
- creating a first attenuating gradient of specular intensity on a first map of the two or more maps, the first attenuating gradient being calculated based upon a first surface reflectivity value; and
- creating a second attenuating gradient of specular intensity on a second map of the two or more maps, the second attenuating gradient being calculated based upon a second surface reflectivity value, which is different from the first surface reflectivity value.
18. The method as in claim 17, wherein calculating the specular reflection of the surface based on the intersection point further comprises:

identifying a first intersection point between the ray and the first map of the two or more maps;

identifying a second intersection point between the ray and the second map of the two or more maps;

5 determining that the surface reflectivity value of the surface within the three-dimensional model is between the first surface reflectivity value and the second surface reflectivity value; and

 calculating the specular reflection of the surface based on the intersection points by interpolating between information stored at the first
10 intersection point and information stored at the second intersection point.

19. The method as in claim 10, further comprising determining specular reflection values for additional points on the surface of the object by interpolation of specular reflection values of two or more object surface points.

15

20. A computer program product for use at a computer system, the computer program product for implementing a method for rendering a specular effect within a three-dimensional model, the computer program product comprising one or more computer storage media having stored thereon computer-executable instructions that,
20 when executed at a processor, cause the computer system to perform the method, including the following:

 creating a map that comprises a representation of at least one light source within a three-dimensional model;

 casting a viewpoint ray to an object surface point, wherein the
25 viewpoint ray comprises a ray extending from a user perspective within the three-dimensional model casting a reflection ray of the viewpoint ray;

 identifying an intersection point between the reflection ray and the map; and

 calculating the specular reflection of the object surface point based on
30 the intersection point.