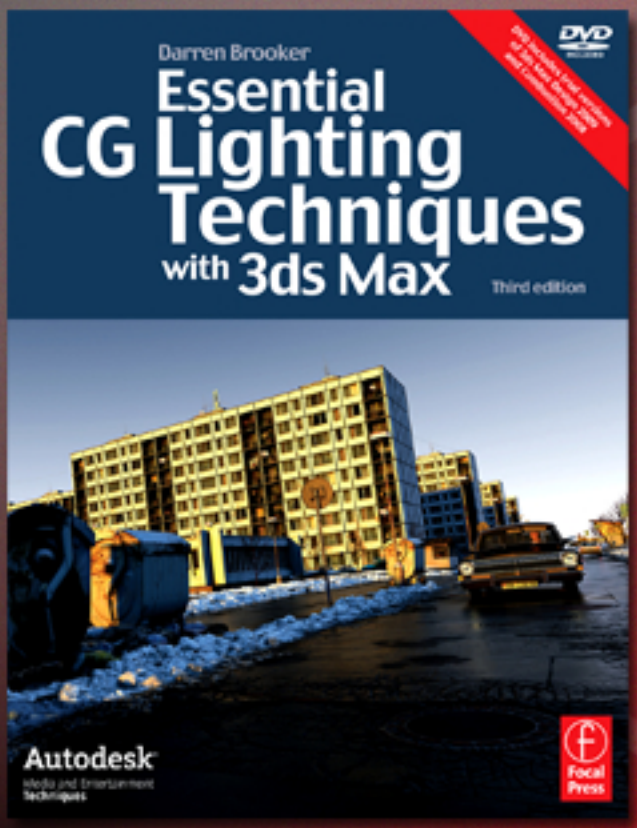
Like what you see?
Buy the book at
www.focalpress.com



Essential CG Lighting Techniques with 3ds Max, 3ed

Brooker

ISBN 978-0-240-52117-6

4

'Envy will merit as its shade pursue,
But like a shadow proves the substance true.'

Alexander Pope

The importance of shadows

Though they might be considered something that things get hidden or lost in, shadows are actually so vital in terms of composition and the role they play in defining spatial relationships, that the importance of shadows in a lighting scheme simply cannot be overstated. The human eye takes a cue from shadows not only in judging where a light source is located, but also what an object is made of, how far away it is and how it relates spatially to its surroundings. Shadows vary enormously in their many qualities of shape and form with the environment's illumination. It is the ability to reproduce these characteristics that is one of the cornerstones of obtaining realistic renderings.

As well as being one of the biggest aesthetic considerations for a lighting artist, in the world of CG, shadows are also one of the most important technical aspects to get to grips with. From the initial choice of algorithm, shadow casting can be a computationally intensive business, so knowing the rules of how to best

Image courtesy of:
Denis Tolkishevsky
www.to3d.ru

represent your lighting scheme's shadows is vital, as is then learning how to break these rules and trick your way to creating convincing shadows in a fraction of the time. Though the ability to hide things in shadows can actually prove very useful, both from a storytelling and a quality control perspective, the visual role that shadows play is more considerable than you might first guess. They serve many purposes visually in terms of composition, detail and tonal range.

Perhaps the most obvious function that a shadow has is to serve as a visual cue to both depth and position. Without shadows, as you can see in Figure 4.01, it's difficult to judge where the different elements are located relative to their environment. The relative size of the various cubes gives you a clue about their depth in the image, but without knowing that the cubes are all of the same size, this cannot be taken as given.

With the shadows present, however, it's easier to judge the positions of the cubes. The size of the different objects becomes apparent and we can see which are actually suspended in space and which are resting on the ground plane.

You can also see from this same image that the shadows help impart a better tonal range to the rendering. This is of particular importance when you are dealing with environments consisting largely of similar colors. Without the shadows, the elements that make up such a scene would be more difficult to tell apart. The contrast that the shadows bring gives us important visual cues which help to shape the space and define the elements within it.

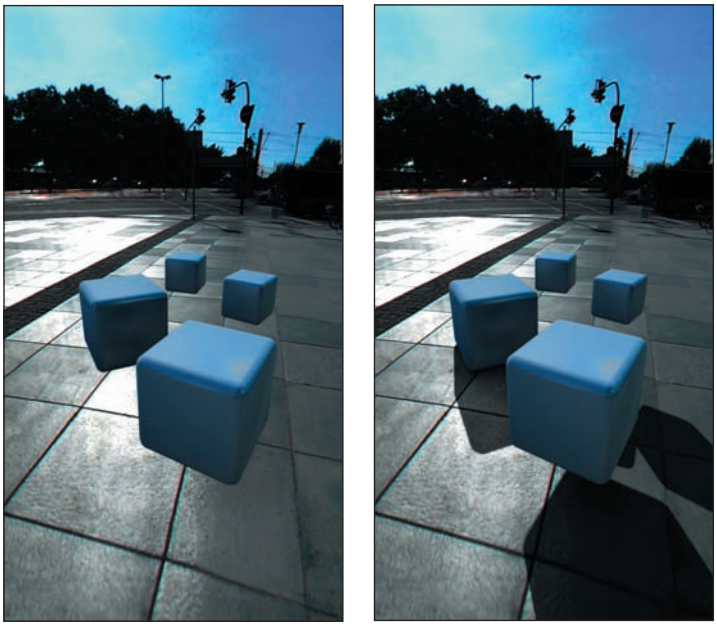


Figure 4.01
Without shadows it's difficult to judge relative positions of objects

The way in which lighting can help show off the modeling work in a scene is of paramount importance. CG productions deal in three-dimensional forms, and though this is invariably presented as a series of flat rendered images, lighting has a pivotal role to play in reinforcing a production's 3D nature. For instance, lighting can be purposefully placed to produce a shadow that reveals the outline lying perpendicular to the camera, which as such would not normally be apparent. Likewise, using a throw pattern such as one that would be produced by a horizontal or vertical blind can be an unusual way of emphasizing a subject's three-dimensional form, which might otherwise have been less apparent.

Furthermore, it's not just what's framed in the image that shadows tell us about, they also give us clues as to what lies outside the rendered frame. The sense of what's going on outside the frame is important in film, and a long creeping shadow can tell us of an approaching figure that reveals a little of the story without giving away the identity of the character. Shadows inform us as to what lies in the space around the viewer, and what is out of shot can go a long way in terms of atmosphere and mood.

As you'll examine in further detail in Chapter 17, it's important to consider the overall composition and balance of your output and shadows can be very useful as a compositional device. They can be used to give detail to large areas that might be otherwise too bare, to frame key elements and to draw your audience into a certain area of the image. Shadows can be an incredibly useful tool in terms of quickly establishing the all-important focal points of a rendering, by obscuring the areas surrounding a focal point; you are effectively framing part of the image.



Figure 4.02
Shadows give us visual clues as to what lies outside of the frame

Motorola *Grand Classics* by Smith & Foulkes at Nexus Productions
www.nexusproductions.com

Additionally, shadows reassure us that objects are sharing the same space. This might not seem a big consideration, but in CG scenes that don't relate to normality are common and anything you can do to suspend your audience's disbelief is helpful. We all know that dinosaurs are extinct, but we are willing to accept their presence when immersed in a movie. However, if one came stomping towards the camera without a shadow tying it into its environment the whole illusion would immediately be shattered.

Shadows help to bring such disparate elements together into a visually cohesive whole. If you want an audience to accept a scene which is somewhat implausible, the shadow can be a great tool in creating convincing interaction that assures the viewer that what they are seeing is actually happening. Without the subtle interplay of shadows, even the most photorealistic of scenes becomes less credible, and the human eye is so used to seeing shadows that even for the most casual observer it does not take much to stretch the illusion too far.

The technical side of shadows

Lighting algorithms can be roughly divided into two categories: direct illumination and Global Illumination. The first method can be best understood by imagining a scene composed of objects and lights. Direct illumination only calculates the light that is received at an object directly from the light source. The lights cast light directly onto the objects, unless there is a further object in the way, in which case shadows are rendered. The light that bounces up off the floor and other surfaces of a scene and back into the environment is not calculated using this model. This way

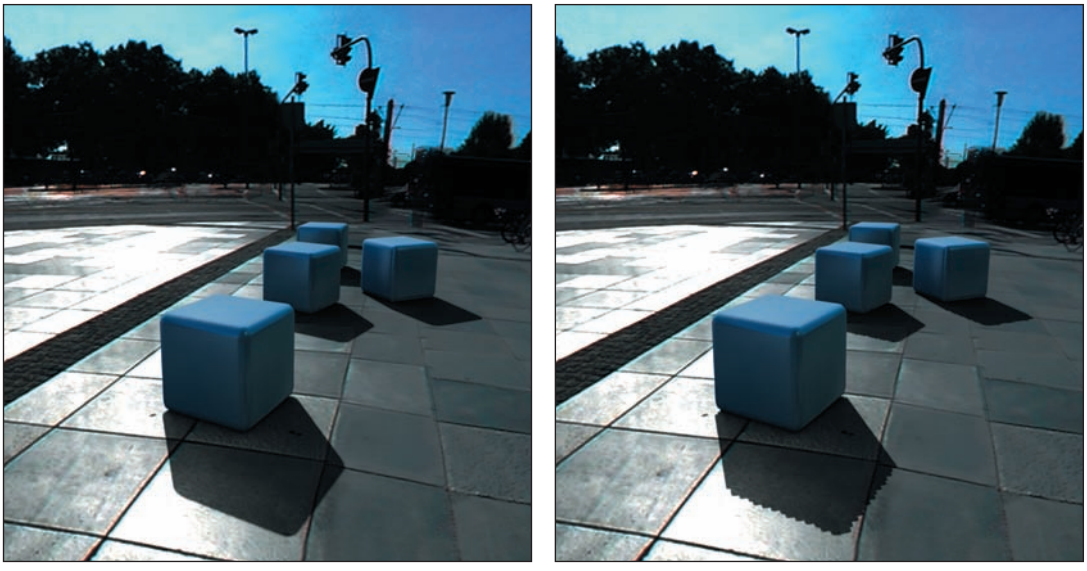


Figure 4.03
Rendering with shadow maps can produce undesirable results



Figure 4.04
Showing the color cast by transparent objects is often only possible using raytraced shadows

of working with direct illumination within 3ds Max would involve using standard lights and the scanline renderer. This has been the principal lighting method used in 3ds Max since its first release, but as of 3ds Max 2009, mental ray has become the default renderer and so Global Illumination has moved up the agenda.

Global Illumination models not just this direct lighting component, but also the indirect light introduced when light hits a surface and bounces back into the scene. If you used direct illumination to render a scene with a single default shadow casting light source, the shadows would be pure black, as would any surface that was not receiving direct light. In real life, objects that are not directly lit are quite visible. Introducing 3ds Max's ambient light will solve this to a certain extent, but as we discovered in the last chapter it will address this by adding a uniform amount of light across a scene. This does not take into account the changes of intensity and color that real-world ambient light demonstrates, and it is here where Global Illumination comes in.

It is this ambient light component that Global Illumination techniques address, and these techniques do so by calculating the many bounces of light around an environment from the scene's different direct light sources. Those familiar with both direct and Global Illumination will know that placing one light will never



Figure 4.05
Shadow maps are more controllable in terms of their softness

give you good results with the former method, yet with the latter, quite beautiful results can be attained with just a single light, as we'll discover when we get to these techniques.

In contrast, working with direct illumination involves the placement of many lights to simulate this bounced ambient light within a scene. Shadows also need to be simulated and in terms of tools there are two basic types of shadow that you have at your disposal: shadow maps and raytraced shadows. Though you might think that two algorithms is not a lot to work with for shadow generation, the amount of variation that can be produced between these two methods is wide, in terms of softness, form, quality, color, and most importantly, render times. The difference between one set-up and another might not be vast in terms of the visual results, but the all-important factor of rendering speed might be vastly different.

Broadly speaking, shadow maps work best with soft shadows, whilst raytraced shadows are good for sharp-edged and accurate shadows. In the world of Global Illumination, raytraced shadows are more common, as they work best with materials that feature transparency and opacity, which are particularly common in design visualization. Light transmitted through, as well as bounced between surfaces featuring such materials is automatically calculated using Global Illumination, whilst with direct illumination, this has to be set up manually. Whilst GI sounds great, the calculation of the light bouncing around the scene's environment is calculated using raytracing-based

algorithms and as such it can be very processor intensive. We'll go into exactly how radiosity works in Chapter 7 and look at mental ray in Chapter 10, but for the moment we'll just look at the two algorithms – shadow maps and raytraced shadows – in isolation.

Shadow-mapped shadows use a bitmap that the renderer generates during a pre-rendering pass of the scene. This bitmap, called a shadow map (or depth map in some applications) is then projected from the direction of the light. 'Depth mapping' is perhaps the more accurate term, as the calculation involves numbers that represent the distances from the light to the scene's shadow casting objects. With this information pre-calculated, the rendering process does not cast light beyond distances specified in this map, making it appear that shadows have been cast.

Raytraced shadows are generated by tracing the path of rays sampled from a light source. This process is called raytracing, and by following the rays in this manner the software is able to calculate to a great degree of accuracy which objects are within the light's zone of illumination and are casting shadows. It was stated earlier that raytraced shadows are best suited to sharp-edged and accurate shadows. Though this is generally true, mental ray area lights use raytraced shadows, and can produce shadows whose softness varies in a beautifully realistic manner.

However, ignoring these special cases and concentrating on the two base algorithms for the moment, it's clear the differences between these two processes are considerable, and the choice of shadow type can have a great effect on both rendering speed and output quality. This is particularly true if you are getting into the render-intensive realm of soft raytraced shadows using area

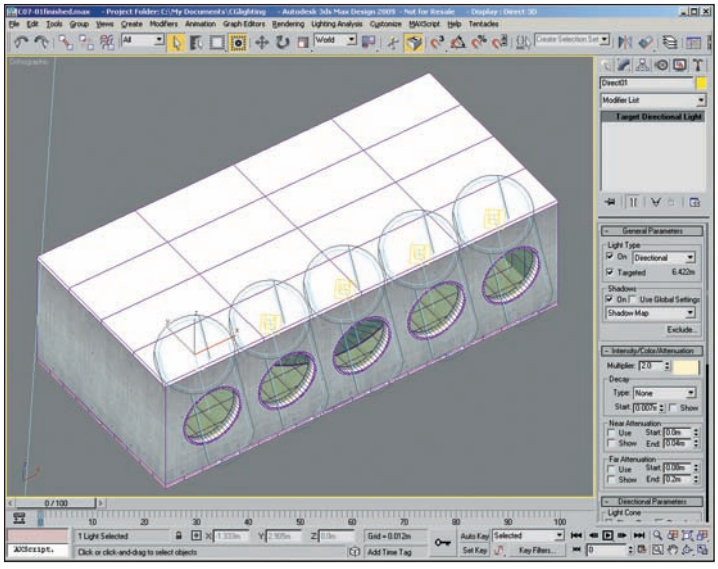


Figure 4.06
Using shadow mapped lights side-by-side can be more efficient



Figure 4.07
Soft raytraced shadows fall off more realistically than shadow maps

Image courtesy of:
© Tobias Dahlén
www.rithuset.se

Illustration for an intranet-solution for Thalamus.
Agency: Mogul Sweden

lights. Shadow mapping is less accurate, but generally requires less calculation time than raytraced shadows. Finally, showing the color cast by transparent objects is only possible using raytracing. Generally, raytraced shadows require little by way of adjustment and they do not generally have swathes of controls, so in terms of fine-tuning, they require less effort. Shadow maps, on the other hand, have far more features and functions, so generally take a fair amount of tweaking from their default settings. The reason that this adjustment is necessary is because shadow-map shadows are only bitmaps, and you need to keep in mind their resolution in relation to your distance from the shadow, and the detail required within the shadow.

If their resolution is too low, the shadow can end up looking blocky and crude. If shadows appear too coarse and jagged when rendered, the map size needs increasing or blurring. However, the greater their size, the more memory required and the longer the shadow takes to generate. If hard edges are required, there can come a point where using raytracing can become a better option, depending on the complexity of the scene. If you have enough RAM to hold the entire scene, including the shadow maps, in memory, then shadows won't affect performance, but if the renderer has to use a virtual

memory swap file, rendering time can slow considerably. The upside of shadow-mapped shadows is that they are much more controllable in terms of their softness, and thus it's easier to control the all-important trade-off between quality of output and render time. One further point of note is that as an omni is the equivalent of six shadow-casting spotlights, the memory requirements of shadow-mapped omni lights jump up as a result, as obviously do the render times if your machine cannot hold the scene and its shadow maps in memory.

Sometimes, only raytraced shadows will do. If you were attempting to render a scene with transparent objects, this method's ability to handle transparency and the transmission of light, can make it the simplest choice, if not the method with the shortest render time. In attempting to produce convincing results with this kind of scene using shadow mapping, you'd have to place additional lights that did not include the transparent surfaces in their shadow casting in an attempt to cheat the correct look. Nevertheless, a convincing cheat that works in half the time of the raytraced solution might be exactly what is needed in a realistic production environment.

The key to controlling render times with shadow mapping depends on several things, the first of which is the shadow's resolution, which dictates the level of detail within the shadow cast. Raising this value makes for increased accuracy, but with the usual penalty on memory requirements. Too low a resolution can result in blocky aliasing around the shadow's edges and the larger the light's coverage, the greater the distance this map has to be spread over, so again, the resolution might need to be increased. By keeping any shadow-mapped lights restricted to as tight an area as possible you are making certain that these shadow maps are used efficiently. After tightening the light's hotspot and falloff values, reduce the shadow map size as much as is possible. When shadows are required across a significant space, it can be much more economical to use an array of lights with smaller shadow maps rather than just one light with a huge shadow map.

As you can see from Table 4.01 on the right, eight lights with a shadow map resolution of 512 placed side-by-side would demand 8Mb of memory to render. This compares favorably with a single light of resolution 4096 (the same as eight lots of 512), which would demand 64Mb, eight times the amount. Alternatively, one light with its shadows turned off could be used to illuminate the entire area, and then shadow casting spotlights could be placed selectively in the various locations where they were needed, thus keeping things as tight and efficient as possible. Furthermore, 3ds Max allows lights to overshoot, which limits shadow casting to within the light's cone, but allows the illumination to overshoot this area, which can be useful for large-scale illumination coupled with efficient shadows over a specific area.

Table 4.01
Shadow map memory requirements

Resolution	RAM
512	1Mb
1024	4Mb
2048	16Mb
4096	64Mb

Shadow map resolution^2 × 4 = memory requirement

Finally, as we've already mentioned, shadow casting omni lights should be avoided wherever possible, as the memory requirements can be up to six times that of a spotlight. To avoid the crisp shadows that are seen all too often in CG, you'll need to acquaint yourself with every aspect of your shadow's controls. Physically small lights do cast hard-edged shadows, as do distant light sources, but in real life the majority of lights cast shadows with a degree of softness. The larger the light fitting, the softer the shadows cast, and whilst shadow mapping can certainly produce soft edges, this is only possible in the scanline renderer when using the Advanced Raytraced shadow type, or when using the previously mentioned mental ray area lights, which we'll look at in Chapter 10. Both of these shadow types can take a prohibitively long time to render in a production environment.

A shadow-mapped shadow is made soft by applying a blur to the shadow map itself. Increasing the amount of blurring increases the render time, but it is worth bearing in mind that smaller resolutions of shadow maps can be used when you are blurring your shadow maps in this way, thus extra time spent computing the blur can result in a saving in the amount of memory that a shadow requires at render time. Blurring raytraced shadows within 3ds Max, as we've already discovered, involves the use of the Advanced Raytraced shadow type or mental ray area lights, but the algorithms behind these methods can see render times increase unrealistically. As you can see in Figure 4.07, the raytraced and advanced raytraced shadow types can produce quite different results. What is worth noting is that both shadow types are capable of hard-edged shadows, but only the advanced raytraced shadow type is capable of soft raytraced shadows.

These soft raytraced shadows are about as good as it gets in terms of quality, but as you might expect this comes at the expense of render time. If you are willing to accept this price, these algorithms can produce beautifully realistic results, with shadows growing softer as the distance from the shadow casting object increases. The fact that these methods can take a prohibitively long time means that their use might be best limited to one-off still renderings. For longer sequences, finding a more workable solution like an array of shadow mapped lights will often be preferable, as this would take a fraction of the render time, and that's why learning how to cheat shadows is what everyday production lighting techniques often come down to.

Faking it

Whilst using the accurate raytraced soft shadow methods will generally give you the best results in the most straightforward manner, the fact that shadow calculation is the most time-consuming part of a light's rendering means that we often need

to fake our way to a quicker solution. There are several tricks and techniques employed in CG to save on render times.
The most basic of these techniques is to use lights with negative brightness to add fake shadows into a scene. By adding such a light with its multiplier or intensity given a negative value, you can selectively darken a region. As we've touched on before, this technique is most commonly employed to selectively and subtly darken a scene in areas like the corners of rooms, but it can also be a useful way of cheating quick rendering soft shadows.

This technique involves targeting a spotlight at the base of the object where the fake shadow is to appear. This light should be set to not cast shadows and the object that is to cast the shadow should be excluded from the light's illumination. Adjusting the Multiplier to a negative number will darken the area at the object's base rather than lighten it. Adjusting the hotspot and falloff will control the falloff of this negative illumination, which will control how soft-edged this 'shadow' appears.

By constantly evaluating different methods of lighting your scenes and minimizing the need for shadows you will certainly keep render times down. Simply underlighting the areas where you want the shadows to fall might sound strange, but can be effective and saves your software a whole lot of bother.

This practice is commonplace when it comes to designing simple light fittings. Rather than simply use an omni light to represent the bulb and the lightshade to cast shadows on the ceiling and floor, two spotlights can instead be used inside the shade – one oriented upwards and one downwards – whose cones fit the circular holes in the shade making it appear that the lightshade is actually casting shadows. A third light should also be introduced in the form of an omni which will illuminate the semi-transparent shade and mimic the light passing through it. By using three lights, none of which is set to cast shadows, in place of a single source that uses shadows, you not only speed up the render, but allow the lamp's illumination to be more closely manipulated.

Shadows-only lights can be used to create shadows without adding any light to a scene, something that obviously is not possible when lighting for film. However, if shadows are the most costly part of rendering light, why would you want to introduce shadows-only lights? The value of these lights is found in terms of cheating lighting angles to hide a modeling imperfection perhaps, or revealing something that would otherwise fall in shadow, or even as a stylistic or compositional device. The use of shadows-only lights also gives you more individual control over the shadow's color, saturation and so on, without having to alter the scene's lighting. Furthermore, as touched on previously, these lights, when used selectively to cast shadows in restricted areas, can provide a very controllable and

efficient solution in large illuminated areas of a scene that can first be lit with a light set not to cast shadows. If you do look in 3ds Max's help files for shadows-only lights, you won't find anything. That's because there is no such thing in 3ds Max (indeed, there is no such thing in most 3D applications) as a shadows-only light. However, this type of light is easy to construct; you first need to create the light that casts the shadows that you desire, and then make a clone of it. Shadow casting should be turned off for this new clone and the multiplier should be set to the same amount as the first, but with a negative value. The result of these two lights is a shadows-only light, with the illumination of the first light counteracted by the negative multiplier of the second. Alternatively, a single light can be used with a Multiplier of 0, a white shadow color and a negative Density value. More about shadows only lights later.

When to fake

Though there are only two widely used algorithms for shadow generation, as this chapter has demonstrated, there are a huge number of ways of manipulating your lights to produce results that don't take an age to render. Knowing when to use a faked solution and when not to fake is something that comes with experience, and your own reasoning will become your best tool.

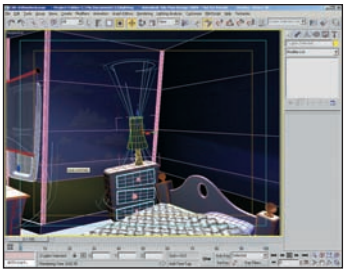


Figure 4.08
By reducing the need for shadows, you can keep render times down

Yes, computers are getting faster all the time so you might be able to get away with using increasingly intensive algorithms, but a lot of the decisions will be based on a studio's production schedules, and whether they allow for the extra time it takes to set up a convincing lighting design using the methods outlined here. Indeed, more often than not you'll be asked to produce results that render as efficiently as possible in little time, due to other production demands. In this situation, it is not viable to set up a



Figure 4.09
Faked solutions that render quickly can be a production demand

quick, accurate solution that is slow to render, nor is it ideal to spend a long time experimenting on a faked solution that renders quickly. In these situations, the only way to make a quick compromise is to know as many ways of completing a lighting task as possible, whether faked or not. Knowing all the options will enable you to best find a happy medium between quality of output, speed of render, set-up time and the degree of control that each method furnishes the scene with.

To use shadows or not?

At render time, casting shadows is the most computationally intensive part of lighting. However, this is not the only reason you should consider restricting shadows within your lighting scheme. Visual considerations can sometimes also dictate that only one or two of your lights need to cast shadows. In considering this, you should ask yourself several questions, some of which might only become apparent as you begin to render and refine your output.

The primary consideration concerns your scene's light sources and what they represent. Does the scene need shadows to fall from any particular source in terms of the necessary realism? With a single shadow casting light in place, the next question you should ask of this source is: is it enough on its own? Single shadow scenes can work well, and a common set-up is to have the key light casting a shadow and all fill lights set not to. For scenes requiring no great complexity of lighting, this is generally a good method if what you desire is a clean, straightforward solution.

However, if you decide that there are other places within the scene that shadows should be being cast from, you should turn these on and work with these shadows in place. Now you are in a good position to judge whether extra shadows are needed. Even with several light sources casting shadows as you might expect them to, things might not look visually cohesive, and you might want to consider adding shadows from one or more fill lights to tie things together. This is possible even if you just have one shadow-casting source, as the ambient light that bounces around the surfaces of a scene can cast additional subtle shadows. Turning on shadow casting for your lights representing this bounced light can help to build up a subtle level of secondary shadowing that really adds to the realism.

Having fill lights that cast shadows is particularly necessary if your key light becomes obstructed within an animation, as it will cast large shadows as a result, and these can look somewhat flat and uniform. Secondary shadowing adds a depth and variation to the image that can impart a sense of life. However, too many shadows can begin to compete, especially when the surrounding environment is very simple, and a subtle approach is more often

than not necessary to avoid disorderly results. Too many light sources casting shadows, especially if these shadows are all coming from different directions, can produce results that are visually distracting. Indeed, look at the world around you, and you can see that most shadows are soft edged and subtle, so a couple of underplayed shadows can often look more convincing. Furthermore, you should not rule out using no shadows whatsoever, if the style you're looking for is abstract or cartoonlike, then the lack of shadows will improve your render times no end and can impart a stylized edge that when done right can be extremely appealing. With the lighting set up to match the environment, the variations in unlit dark areas that appear on the subject can appear convincing enough, and the need for actual shadows is not always necessary.

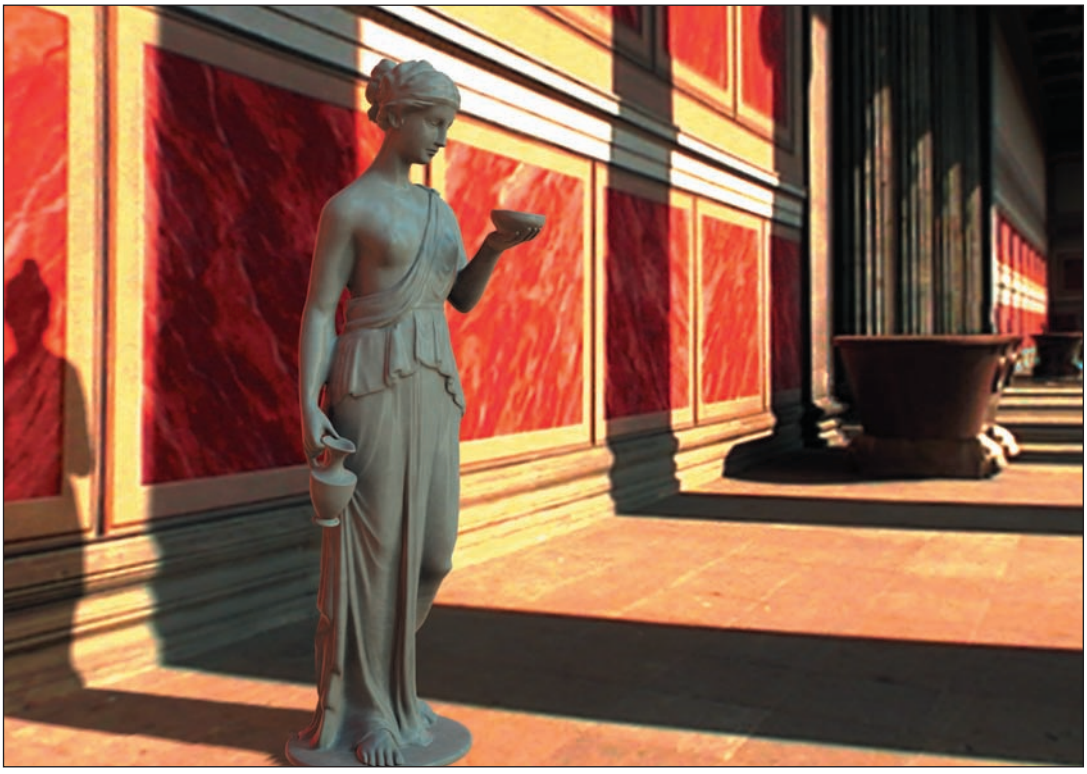
Figure 4.10 (top)
Having no shadows at all is most times an unrealistic proposition

Figure 4.11 (bottom)
Shadows play a key part in tying a CG element into a scene



Shadow saturation

The saturation of shadows is a major factor in achieving realistic and believable results, and whilst shadows that are too light can look just as unrealistic, it's generally darkness that causes the problems. Even when a lone bright light source is illuminating a scene, the light bouncing off the surfaces of the environment lightens the shadow it casts. Adjusting the saturation of your

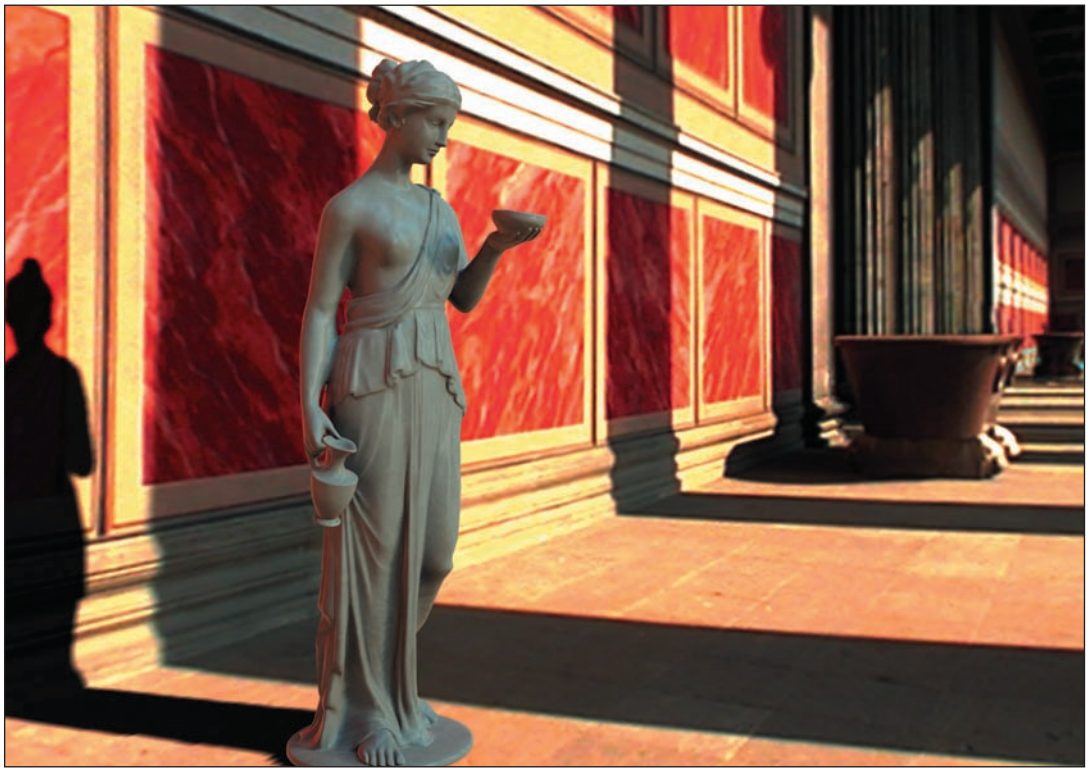


shadows is possible in several ways. The first and most obvious method is by using the Shadow Color and Density controls. However, when using this method you must be careful that the cast shadow does not start to look too different from the unlit portion of the object casting the shadow. This can result in an unrealistic amount of contrast between these two adjacent areas. The unappealing second option is to use global Ambient setting to brighten the whole scene slightly, but as discussed in the previous chapter, this control should be avoided at all costs. Allowing even a small amount of ambient light into your scene deprives you of the full range of tones available, as pure black will now appear slightly grayed. Furthermore, the ambient light that is added will not bring out any details on the object's unlit side, as ambient light simply adds an unvarying amount of illumination across the scene.

The only real way to lighten your shadows is to add fill lights to recreate the light bouncing off the surfaces surrounding your object. This lightens the cast shadow and the unlit portion of the object consistently, and the unlit portion is illuminated in a way that brings out any detail across this area. Fill lights representing the light bouncing off the surrounding surfaces that are colored to match their material, impart not only a sense of realism, but add to the all-important visual richness and depth of a rendering.

Figures 4.12 & 4.13
Shadow saturation is one thing that needs to be watched closely









part 2 > techniques

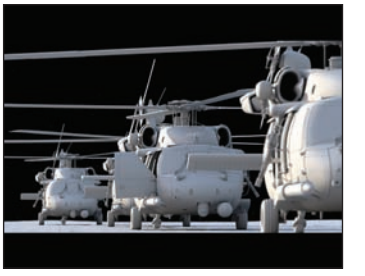


Image courtesy of:
Andre Cantarel
www.cantarel.de

