Introduction to 3DSMax

TNGD25 Lab 5/6

Matte compositing

Contents

1 Mixing synthetic images and photos

Up until now, you have been doing purely synthetic renderings in 3dsMax, where all objects, all materials and all light sources are created in the virtual scene. This is a common and very useful application of 3-D graphics. However, another common application is to mix photos with rendered content. Placing virtual objects in real scenes is comparably easy, so we will focus on that in this presentation. Using the same principles, you can also do the opposite: put real world objects into a virtual environment, but it's a bit more tricky. You need to plan your shots vcarefully so that the real world object or actor can be separated from the background. A good tool for this is a green screen studio, but we don't have access to one. A plain coloured wall is sufficient in many cases, but let's not go there. This document describes how to place a virtual object in a photo of a real world scene.

Large parts of this document is very similar to an official 3dsMax tutorial about the *Matte/Shadow/Reflection Material*, although we are using the *Compact Material Editor*, and we are going to treat light and shadows properly.

2 Compositing

Mixing images from two sources, e.g. 3-D renderings and photos, is called "compositing", and it's a very important tool for non-3-D special effects (SFX) as well as modern 3-D graphics ("visual effects", VFX).

To do it, you need the two images you want to mix, and a so-called *Alpha mask* to determine which image should be used for which pixel. Rendered images from 3dsMax of objects against an empty background have an inherent alpha mask which can be saved with the image. The mask can also be created in Photoshop, even though semi-transparent objects would present a challenge there.

Two images, a mask image and some image editing software is all that's needed to place 3-D virtual content into real world scenes. However, to make a high quality composite image, you need to carefully consider two aspects of the real world photo when you make the computer generated content: the illumination and the camera angle. Both must be matched reasonably closely to make the image believable to a human observer.

3 Background image

As a simple but instructive example, we choose to put the classic icon of 3-D graphics, the Utah teapot, in an indoor environment. The photo where we are going to place the teapot is shown in Figure 1.

Figure 1: A simple real world scene where we are going to put our synthetic object.

4 Light probes, HDR and IBL

The indoor image in Figure 1 has a complex, soft illumination with a lot of indirect light, and a fairly bright sun behind some thin clouds and a curtain. The scene has a lot of complex illumination, with a lot of secondary light. The scene has a complex illumination but a predominantly neutral color of the light. It's pretty difficult to recreate this illumination using the standard tools in 3dsMax, so let's use a technique called Image Based Lighting (IBL). The general idea is to capture a panoramic image of the environment, from the vantage point of the object that will be rendered, and use the pixel values in that image to determine how much light, and of which color, that would shine on a virtual object placed at the center of the panorama. To acquire an image like that without spending lots of time on panorama stitching, it is very nice to have some sort of wide angle optics.

There are suitable fish-eye lenses with 180 degrees field of view, but a more common approach is to capture an image of a reflective sphere, e.g. a polished steel ball. In the image of the ball, you see almost every direction from the position of the ball, except for a small region directly behind the ball and another small region behind the camera. As long as those particular regions don't contain any strong light sources, the image is suitable for IBL as a*Light Probe*.

In addition to capturing all directions, the image used for IBL must not have any over-exposed pixels. Direct images of light sources are thousands of times brighter than the area they are iluminating.

Capturing a dynamic range that large requires High Dynamic Range (HDR) photography. There are cameras that claim to have HDR capture built in, but the technology is immature, and you can use an ordinary camera to get the HDR image. In that case, you capture several images spanning the full dynamic range of the scene, and use software to assemble the HDR image afterwards. A modern digital SLR camera can vary the exposure within a wide range, and is therefore very well suited for HDR capture.

The 12 images we took to make the single HDR light probe for this exercise are shown in figur 2. They are captured in a manner such that the brightest spot is not over-exposed in the darkest image, and only the darkest parts of the image are not over-exposed in the brightest image. Between these extremes, trhe shutter speed was changed in steps of 2, from 1/1000 second to 2 seconds,

Figure 2: 12 Images of a reflective ball, to capture the illumination

To assemble the images into one single image with high dynamic rang, you can actually use Photoshop. Exactly how to do it depends on the version of the software, but the functionality has been around for a decade. Exactly how to do it depends on which version of Photoshop you are using, but there are good instructions in the Help section in Photoshop, under the keyword "HDR".

When assembling the HDR image, keep in mind that it will only be used for lighting and reflections, and therefore it usually doesn't need to be very big. The HDR assembly is a complicated task, and it will be sped up considerably if you use smaller images.

A final HDR composite of the mirror sphere from the scene, cropped to a square image with the edges of the sphere just touching the image edges and a few hundred pixels across, is shown in Figure 3. This kind of image is called a *Light Probe*.

For a good fit with Nvidia Mental Ray, the light probe should be taken in the same camera angle as the background image for the compositing. It's not an absolute requirement, but it makes things a lot easier.

It's worth mentioning that it's not always easy to make a good light probe. The sun might actually be too bright to be viewed straight on and get good pixel values. If the sun hadn't been behind the curtains, even shorter exposure times would have been required, short enough to disqualify many cameras from the task. In that case, one solution is to use a neutral density filter for the camera. Another, more ugly and temporary solution, is to pull the lens slightly out of focus, which makes direct reflections of light sources more blurry and larger, and hence have lower intensity. If all else fails, it's also possible to retouch the HDR image to remove the Sun altogether from the light probe and replace it with a directional light source in the scene.

Figure 3: Low resolution HDR image (32 bits per color channel) of the mirror ball. The image was converted to 8 bits per channel to be printable.

5 The Scene

Start with an empty scene in 3dsMax. Choose *NVIDIA mental ray* as your renderer in *Render Setup*.

To know what you are doing when dealing with compositing, we want to work in real units. In the menu, choose *Customize* → *Units Setup*, and pick centimeter for your units. Create a (*Teapot*) whhích is 16 cm diameter (8 cm radius) and put a boring, gray material on it for now.

Click the little plus sign at the top left of your Perspective viewport. Choose *Configure Viewports...*, find the tab *Background* and select the alternative *Use Files...*. Pick your background image by clicking the button *File...* and find the file `backdrop.jpg`. Make sure that you under the heading *Aspect Ratio* pick the alternative *Match Bitmap* so you don't accidentally change the proportions of the background image.

Zoom in to place the teapot at about the right location in the background, and turn the view to make it face in the desired direction. Render.

Unfortunately, the teapot is rendered against a black background. There is an alpha mask, so you can still do compositing in Photoshop, but a better preview in 3dsMax would be useful. The renderer can also perform the compositing, even though it's not as flexible as doing it in Photoshop as a post-processing step.

In the menu, click *Rendering* → *Environment*. Under *Environment Map*, pick a new map of type *Environment/Background Switcher*. (Why we do it like this instead of simply picking a *Bitmap* directly will be revealed in the next section.)

Open *Compact Material Editor* and drag the newly created map of the tyoe *Environment/Background Switcher* to one of the sample slots. Pick the option *Instance*, not *Copy*, so that the changes you make in the editor also affect the object which has the material on it. Click the map in the Material Editor to see its settings. There are two slots for sub-maps: one for *Background* and one for *Environment/Reflections*. For *Background*, pick a new map of type *Environment/Background Camera Map*, click the button next to *Map* and pick `backdrop.jpg`.

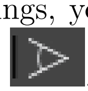
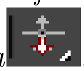
If you render now, the background image will show where there was previously black. TO see the entire image, you need to set the resoluition in *Render Setup* (the button in the *toolbar* at the top) to match the proportions of the background image (2:3). Don't use the full size of the backdrop, just keep the proportions. Specify e.g. 500x750 pixels to get reasonably quick test renderings.

The remaining problems this image are the camera angle and the lighting. We will fix this in the upcoming sections.

6 Camera Matching

To match the camera angle, a task which is formally called *Camera Matching*, you can use automatic methods to compute the position of the camera based on points in the scene, *markers*, whose relative positions are known. For movie sequences with camera motion, this saves a lot of work, bit for still images it can often be just as easy to match the camera position manually. The precision will be lower, but in still images, it's mostly enough with an approximate camera matching. That is what we will do here.

We need a reference object which exists both in our synthetic scene and in reality. The top of the piedestal is a marble slate which measures 25 $\times$ 25 $\times$ 2 cm. Create a *Box*, or even better a suitably rounded *ChamferBox*, with those dimensions and place it in the middle of the work area in the Perspective view. Select the object with the tool *Select and Move*, and key in the xyz coordinates 0,0,0 at the bottom of the main window in case you want to place this exactly in the middle.

Rotate and move the view to watch the box from more or less the correct direction. The box is shown with a too strong perspective distortion. This is because the *Perspective* view has a too large field of view (FOV), and the camera is actually too close. Instead of messing around with the generic Perspective view, let's create a proper camera. You can create a camera from the current view by pressing *ctrl-C rightarrow Cameras rightarrow Create Physical Camera From View.* Now you should have a better overview and control over what the camera sees in the scene. Among other things, you can animate it, you can now set the zoom for it with the button *Field Of View* , and you can move the camera closer or farther away with *Dolly Camera*. (Note that zoom and dolly are two separate operations, with different effects on the image.)

Change *Field Of View*to zoom in a little bit more, pan and dolly the camera until the box is about the right size again, and pan and rotate the camera to view the box from the right angle. Repeat, fine tune and possibly nudge a little back and forth to get the perspective right, but don't spend too much time on it.

To adjust only the perspective without zooming in or out, there is the very useful button *Perspective* . With some tweaking and a little patience, you should be able to find a good match between your virtual box and your real object. Try your best to make the match reasonably good, because the quality of the final output depends quite heavily on this. The scene is simple and not very detailed, but make sure that at least the right edge of the piedestal matches, because of where the shadows will fall in the rendering. Display the objects as *Wireframe Override* while you are tweaking the camera, and you will see better what you are doing.

Put a boring plain gray material also on the box, and render. The box hides the pedestal in the final image, which is not what we want. However, we will rsolve that problem later. We'll get back to this. Now that the camera angle is right, we will fix the ighting.

7 The light source

Because we have an HDR image of a reflective ball, a *light probe*, we will use it for the illumination. There's an empty map slot in the*Environment/Background Switcher*, and that's where our light probe fits. In the sub-map *Environment/Reflection*, pick a new map of type *Environment Probe/Chrome Ball*, and then under *Chrome/Mirror Ball Image*, pick the file "probe.hdr".

Now we have added an environment map that can make shiny virtual objects reflect real objects, thus making them believable as part of the real world scene. To use it, add a *Reflection map* of the type *Raytrace* on the box and the teapot, and test render. The mirror reflections in the teapot should now show its entire surroundings: reflections of its own handle and spout, the box it sits on *and* the environment from the photo of the mirror sphere. Set the intensity of your Reflection map to something reasonably small, like 30% instead of the default 100%, and consider making the teapot a little darker in the diffuse color, to make the reflections show more clearly.

The illumination is still wrong. Image Based Lighitng in Mental Ray is done through a *Skylight*, so create one if those in your scene. In the Modify tab for the *SKylight*, under *Sky Color*, pick *Use Scene Environment*. Now, the intensity and color of the light from all directions will be read from the mirror ball image isntead of neing constant in all directions.

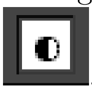
Test render. The teapot is way too dark. This is necause we haven't calibrated our HDR image. It has no idea how much light that really came from that window. (If we had made another HDR image using the same techniques, that one could just as easily have made the images appear way too bright.)

For good correspondence between the lighting in the real world scene and the rendered image, we must change the *Multiplier* for the map *Environment Probe/Chrome Ball* which holds the imag eof the mirror sphere. Increase *Multiplier* to somewhere around 10 to 12 or something. The exact value is not all that important, but there are ways of doing a proper calibration with reasonably simple means if needed. HDR images found in the wild (published on the Internet) are actually very seldom calibrated, so you always have to add that extra step of guessing which *Multiplier* to use.

8 Shadows

The box beneath the teapot is opaque and covers tha background. We don't want this. THe box should be transparent, but we still want the shadows from the teapot onto the box to show on the pedestal in the final image. For this purpose, there is a special material called *Matte/Shadow/Reflection*. Create a material of that type in the material editor and apply it to the box. Render again. Unfortunately, the box becomes black. This is because we didn't tell the material what is supposed to show through it.

In the material *Matte/Shadow/Reflection* thereäs a map called *Camera Mapped Background*. That map should be the same map as the one you are using for your background. Navigate to your *Environment/Background Switcher* map, *right click* on the button *Background map* and pick *Copy*. In an available sample slot int he material editor, change the type of the sample material to *Matte/Shadow/Reflection*. In the settings for that material, right click the button next to *Camera Mapped Background* and pick *Paste (Instance)*. Rendera again.

Now the background image shows through the box, but the shadows make the background darker where they fall on the "invisible" box. This involves compositing semi-transparent black pixels onto the background. Watch the alpha channel in the rendering window by pressign the button , and you will see a little more clearly that this actually works.

9 Reflection

The marble surface is rather shiny in realiyy, so it would be nice if it showed a slight reflection of the teapot in the final image. The material *Matte/Shadow/Reflection* can simulate that as well, as its name says. In the settings for *Matte/Shadow/Reflection*, under *Reflection*, check *Receive Reflections* and pick a medium dark gray color for the reflection to make it reasonably subtle. White give you a mirror-like reflection, gray makes it look more like a polished or lacquered surface.

If you also put a slight bump map on your matte-object (the box), the reflections will be wobbly. FOr our marble surface, that is not what we want, but it's very useful in other situations, like if you want to simulate the reflection of a synthetic object in a water surface.

Tweak the material for your teapot and your box until you are happy with the result. Instead of making your own materoial for the teapot, you can experiment with the predefined materials, like *Autodesk Ceramic* or *Autodesk Metal*. Render a final image in reasonably high resolution and save it. COnsider saving the image not just as JPG, but also as PNG, and open the ÅNG in Photoshop to see that the transparency in the background is saved in the PNG format. You can do your compositing by using two layers in Photoshop if you like. That is often the way professionals do it, because you get an extra opportunity to adjust the color balance and contrast between the two layers. You might also want to add some blur and possibly noise to either of the images to make them match also in terms of fuzziness Compositing directly in the renderer like we did here is actually not very common. Even if you trake care to make everything right in the 3-D production, you still want the options for manual tweaking.

Figure 4: The final composite image, rendered in high resolution.