Real-World Measurements for Call of Duty: Advanced Warfare

by Danny Chan, Sledgehammer Games / Activision Blizzard

1 Introduction

Our goal for the rendering in *COD: AW* was to take a small step towards photorealism. Real-world measurements ended up being an integral part of our understanding of materials and lighting. We recorded many measurements, and while a lot of these were used directly in the game, much of the benefit of these processes lay in gaining an intuition for appropriate values for diffuse albedo, gloss and luminance.

To start off, real-world reflectance doesn't always conform to our expectations (Figure 1). This is a scene with a very bright reflective surface. Normally, we'd expect this kind of reflectance from a smooth, shiny surface with a light source at a grazing angle. But concrete is a material we'd consider rough and the sun in this case is high in the sky. If we were to imagine this scene, we probably wouldn't imagine the bright concrete. This is a photograph that confounds our expectations, yet when we see it, we recognize it as *true*[1]. Getting these kinds of details right was a critical goal for us.

Figure 1: An example of bright specular reflection that confounds our expectations.

[1] In the sense of looking real, or matching "ground truth".

2 Camera Exposure Controls

Before we discuss how we use real-world measurements, we're going to review some basic camera controls since they relate to how we control the in-game camera and how we use photographic techniques to measure diffuse albedo and lighting.

In a camera there are three main controls over how the film or sensor will respond to the light from a scene:

Shutter speed: film exposure time

Aperture: size of the opening through which light travels (how much light is let in within a given time)

ISO: the sensitivity of the sensor

For our purposes in rendering, we're going to simplify these down and combine all these controls into one called *Exposure Value*, which we'll discuss later (Section 3.1).

2.1 Shutter Speed

Shutter speed is straightforward: if you double the time that the shutter is open, you double the amount of light hitting the sensor. There are many situations where we are required to vary the exposure when photographing a scene. For example, *bracketing* a shot means taking a graduated series of exposures. Later, we will discuss varying shutter speed as a method of bracketing when performing our captures (Section 8.1).

2.2 Aperture

The aperture is the opening that allows light to pass through to the sensor. Commonly, we hear aperture size described as $f/2$: "f over two" or "f two". This defines the aperture diameter, D, in terms of focal length, f, divided by an *f-number*, N [Wika]. So $f/2$ means the aperture diameter is half the focal length. If we compute the aperture area from $D = f/N$, we can see that $f/2$ lets in half as much light as $f/1.4$:

$$Area = \pi \left(\frac{f}{2N} \right)^2. \tag{1}$$

Each halving of light is a scaling of N by $\sqrt{2}$, therefore $f/2.8$ lets in half as much light as $f/2$, and $f/4$ lets in half as much light as $f/2.8$.

2.3 ISO Speed for Digital Cameras (ISO 12232)

Digital cameras measure incoming light at its sensor and convert to linear RAW values. For any given scene, the sensor gain can be adjusted to result in smaller or larger RAW values. Then, these linear RAW values are converted to sRGB. ISO setting, or Exposure Index (EI) rating, encompasses both of these steps and defines how the luminance values in the scene are translated to sRGB values.

Oversimplifying a bit, you can think of ISO speed as sensor sensitivity. Generally speaking, doubling ISO will double your sensor sensitivity. In the same way you can double the time the shutter is open to allow more light to hit the sensor, you can double the ISO to double the gain on the sensor to result in a brighter image.

The ISO 12232:2006 standard [Wike] gives digital camera manufacturers a choice in how to define ISO settings for their cameras. We're only going to discuss the two that are more commonly used in newer digital cameras.

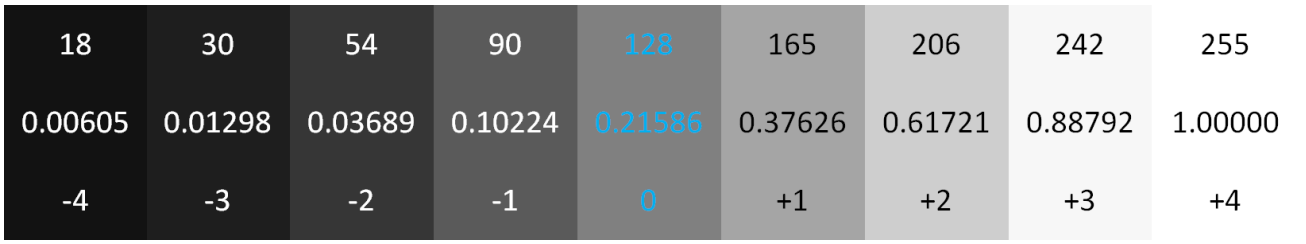

Figure 2: JPEG values from nine separate, bracketed exposures of an evenly lit projector screen surface, taken using a Nikon D800. Top row is sRGB, middle row is display linear, and bottom row is exposure compensation.

Recommended Exposure Index (REI) allows the manufacturer to basically define ISO however they want in order to produce a well-exposed image given the ISO setting. This gives manufacturers a lot of interpretive leeway when determining exposure using multi-zone metering. In fact, we suspect some cameras adjust exposure based on some of the same heuristics we use in-game (Section 7).

Standard Output Sensitivity (SOS) specifies that the average level of the sRGB image must be within $\frac{1}{3}$ stop of 18% gray (commonly referred to as *middle gray*) when the camera is set to auto-expose and there is no exposure compensation [Wike].

We wanted to measure the sRGB values produced by a digital camera using different exposures so we performed a quick test by setting a Nikon D800 to center spot metering and taking bracketed shots of an evenly illuminated projector screen. We bracketed using exposure compensation (Section 3.3) through varying shutter speed (Figure 2). Notice the center exposure, which has no exposure compensation, results in an sRGB gray value of 128, which is 21.5% gray. The linear values in this case refer to display linear values, i.e. post tonemapping, and not to linear input radiance. This is why neighboring exposures, which are one stop apart, are not a doubling or halving of the display linear values.

Generally speaking, when spot metering, we expect cameras to follow Standard Output Sensitivity, which is to say setting exposure based on a metered surface will make that surface appear middle gray.

2.4 Luminance

While the photons that bounce around a scene contribute to the radiometric energy that reach our eyes, what we are primarily concerned with are photometric quantities, which are tied to our perception of this energy. Luminance is a perceptual, photometric value that we consider equivalent to *absolute brightness*. It is this absolute brightness that we tune to and target in our HDR rendering.

When we talk about linear luminance, instead of using $\frac{cd}{m^2}$, we use the equivalent term *nits* since it's easier to say and write. So a bright LCD monitor may be 200 nits, or a light may be several *kilonits*.

3 Exposure Value (EV)

EV is the common currency when talking about lighting. It is the single most important concept of our lighting pipeline. There are three ways we use EVs: as a single exposure control, as a measure of luminance, and as an exposure adjustment.

3.1 EV as a single exposure control

EV is the single combined camera control that determines exposure. We are going to boil down the three camera controls discussed earlier—ISO, shutter speed and aperture—into one. In Figure 3, there are three images captured using different combinations of camera settings that all resulted in the same image.

Figure 3: Different ISO, shutter speed and aperture combinations can equate to the same EV_{100}, resulting in the same image.

Traditionally, EV is an indicator of two camera settings, f-number and shutter speed, independent of ISO [Wikd]. We follow the convention used by our Pentax spot meter, which treats f-number and shutter speed as relative to ISO 100; EV in this case is denoted as EV_{100}. In this way, we have a single value that encodes effective exposure as determined by shutter speed, aperture and ISO. In all three photographs, the effective exposure, EV_{100}, is 14. Whenever we discuss EV as exposure control, we implicitly mean EV_{100}.

Notice the middle photograph has a shutter speed that is twice as fast as the top photograph, but has an aperture area which is twice as large, allowing the same amount of light to hit the sensor. Aside from depth-of-field differences, this results in the same captured image.

3.2 EV as measure of luminance

There are simple formulas to relate luminance to EV [Wikd]. Unfortunately, these formulas are dependent on the brand of spot meter you are using to measure EV:

Nikon, Canon and Sekonic

$$L = 2^{EV-3} \tag{2}$$

Pentax and Minolta

$$L = 2^{EV-2.84} \tag{3}$$

The difference of $\frac{1}{6}$ EV between the different brands of spot meters is the result of different calibration constants [Wikg].

Notice that each difference of 1 EV is a doubling or halving of light. We use EV as a measure of luminance, or absolute brightness. Given an EV, we can convert it to luminance and vice versa—they can be used interchangeably.

Our in-game spot meter matches our real-world Pentax spot meter in measurement, so lighters can match their real-world measurements in-game.

3.3 EV differences as exposure adjustment

EV differences are used to describe exposure adjustments (commonly referred to as exposure *compensation* on digital cameras). An exposure adjustment of +1 EV can be accomplished by doubling the shutter time, doubling the aperture area, e.g. f/2 to f/1.4, or doubling the ISO. Notice however, when adjusting exposure by +1 EV, we are actually *decreasing* by 1 the EV used as exposure control, resulting in a *brighter* image (Section 3.1).

4 Lighting Measurement

There are three metering tools that we use to calibrate the game to real-world values: a real-world spot meter, an in-game spot meter, and an in-tool incident light meter.

4.1 Spot meter

The real-world spot meter is used by lighters to gather *reference EVs* for everything from light bulbs to lit billboards to representative surfaces. It's simple in operation: you just look through the viewfinder, point and pull the trigger and the spot meter measures the luminance across the 1-degree center circle and converts into EV (Section 3.2).

Light bulbs and billboards that are represented visually by emissive materials in-game can be tuned to match their real-world counterparts. We use a scale parameter in the emissive material that is proportional to kilonits. While we use kilonits in this case, moving forward it makes more sense to specify in EV, which is closer to perceptually linear.

4.2 Incident light meter

We provide an incident light meter in the world building tool. It measures the luminance of a 100% white surface that is directly facing the light at the position of the meter. This is different from a real-world incident light meter which measures for the luminance of an 18% gray surface and also takes into account total illuminance, i.e. other light sources. Our light meter only measures the contribution from the paired light, which is an unnecessary limitation. We're considering measuring all incoming light and changing to the convention of 18% gray for our next project to be consistent with real-world incident light meters.

Typically, a lighter may pair every light with a meter and place that meter at the targeted spot of the light. This is mainly to give ballpark figures so that light intensities can be roughly tuned. Lighters follow up with finer spot meter tuning if necessary.

4.3 Colorcheckers

We make liberal use of colorcheckers during real-world capture and in-game measurement. When performing sky capture, we place colorcheckers in the scene alongside a whiteboard. The whiteboard is used to record gray square luminances in EV, both in light and in shadow. In this way, we don't need to keep track of separate meta-data related to the sky capture. Since the colorchecker and whiteboard are placed on the ground below the camera, the environment geometry will cover up that portion of the skybox during normal gameplay. We also place colorcheckers in-game so we can spot meter the gray squares and match against any real-world measurements (Figure 4).

4.4 Light vs. Shadow

Probably the most important ratio in lighting is the EV difference between lit and shadowed areas. Getting this *relative* difference correct is more important than getting accurate absolute values. Where

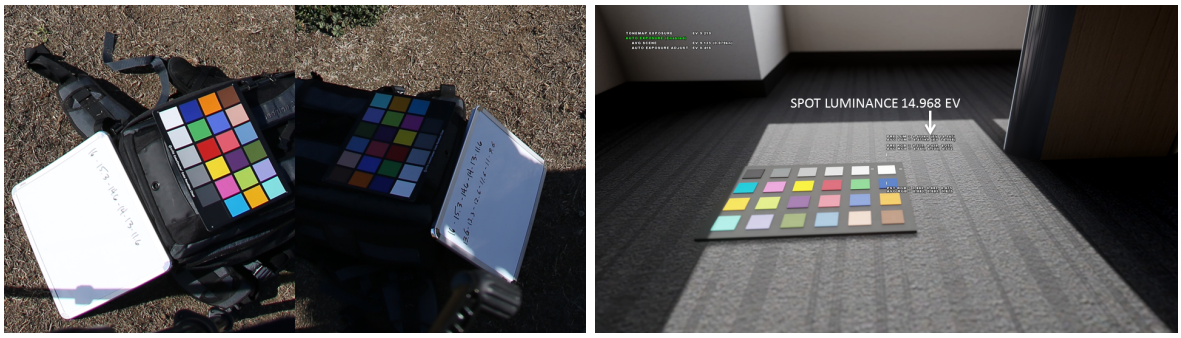

Figure 4: Colorchecker and whiteboard in sky capture photographs (left). In-game colorchecker being measured using in-game spot meter (right).

you measure the shadow EV is important, especially outdoors, since the shadow at the base of an object (e.g. near the roots of a tree) is darker — there is less of the sky visible from that point. As a first step, lighters normally measure lit and neighboring shadowed areas. Then, for example, in an outdoor scene, sunlight and sky intensity can be tuned to achieve a target EV difference.

Since our sky capture process includes recording light and shadow EV values, we can also achieve accurate *absolute* EV values in our rendered scene fairly simply. Our sky captures (Section 11), once processed into HDR, are linear but at an arbitrary scale. We scale our skyboxes in-game such that the colorchecker gray squares in the skybox match their recorded EVs. Once the gray squares in light are tuned to hit their targets, the gray squares in shadow should also match their recorded EVs[2].

In order to tune the punctual sun light source in our rendered scene, we place an in-game version of the colorchecker into the scene and scale the sun intensity until the in-game gray squares match their recorded EV values.

4.5 Lumens and world scale

We provide the ability to specify *lumens* for our lights. The idea behind this is we'd be able to look up lumen output for a light bulb and apply that in-game. However, because of our camera field of view and player movement, our game worlds have a much larger scale than the real world. If we were to use real-world lumens, our game world would be lit too dark. No single scale value can be used to adjust lumens since the world scaling is non-uniform, e.g. doors may be twice as wide but only one and half times as tall. Real-world lumens, then, are not very useful to us and are never used. Instead, lights are tuned such that representative surfaces in the game world hit target EVs. This is an important point: we only pursue physical realism so long as it's possible.

5 Lightmap Validation through Measurement

Having an in-game spot meter proved to be extremely useful as it helped us to debug some significant lighting issues. In early testing of our whiteboxes, we noticed that the lighting was wrong (Figure 5) and discovered the in-game measurements of surface brightness did not match our real-world measurements. We found that adjusting our lightmap baking tool by scaling the amount of light reflected per bounce by 0.3× gave us *approximately* correct lighting values (Figure 6).

You might notice that 0.3 is around $\frac{1}{4}$ or $\frac{1}{\pi}$; in other words, we were off by a factor of 4 or π somewhere in our lighting calculations. Both of these factors are pretty common when performing integrations over solid angles, but we were unable to find the issue in these areas of the code. For several months we left this 0.3× scale factor in since it gave us reasonable lighting values. It was only

[2]This is also a method of verifying linearity in the HDR skybox.

Figure 5: Lightmaps are incorrect.

Figure 6: Lightmaps are correct.

Figure 7: The range of luminances in real-world scenes (in *nits*).

later that a bug was discovered in our supersampling code that effectively scaled our bounce energy by 4×! Once this bug was fixed, we were also able to remove the scale factor; the difference between 0.3 and 0.25 was not enough to change the look nor the measurements noticeably. This was a validation of the lightmap work we had done thus far—we closed the loop since our measurements and lightmap calculations now agreed with one another.

6 High Dynamic Range

The visible world spans a large range of luminances: from starlit scenes around 0.0001 nits to sunlit scenes around 4000 nits (Figure 7). The sun itself is around 1.6 billion nits. The HDR buffer we render to can only represent a fraction of these visible luminances so we provide a per-level tuning parameter to scale real-world luminances into scenebuffer values.

6.1 Scene referred vs. display referred

It's important to make a distinction between scene-referred values and display-referred values. In our discussion, scene-referred values encode the real-world (or in-game) HDR value, while display-referred values are the LDR values we map to the screen. When we talk about a scene, it means we are either talking about the real-world or the values rendered into our HDR buffer.

It's this mapping between the *scene HDR* value and the *display LDR* value that is the result of exposure and tonemapping.

6.2 Exposure and Tonemapping

Although exposure and tonemapping work in concert, it's important to keep these two concepts separate. Given a scene, we can think of exposure as *shifting* our values into range, while tonemapping can be thought of as *compressing* those values to fit on the display.

Figure 8: A dark (bottom) and bright (top) exposure for a night (left) and day (right) scene. The night scene averages 2 EV, while the day scene averages 14.3 EV. Notice the bottom left and top right images are most appropriate.

Since tonemapping and tone curves are covered in detail elsewhere, we'll just mention that we started with Reinhard and eventually moved to an S-curve because of authoring implications (Section 9.4).

7 Exposure Determination

Achieving proper, precise exposure of a scene is often overlooked, but exposure is important for color perception as well as artistic intent. At a broad scale, exposure determines how bright your display is; at a finer scale, exposure allows targeting of light or shadow for specific output brightness.

7.1 Subjective Interpretation

How to expose a scene is a matter of subjective interpretation. In the early days of HDR in games, exposure was usually *perfectly* adaptive: the display averaged to middle gray, regardless of the scene lighting conditions. This is equivalent to using no exposure compensation (EV 0) on a camera set to auto-expose.

Photographers don't capture scenes this way. As a general rule, night scenes are exposed so they appear dark, and day scenes bright (Figure 8). Display brightness is generally proportional to scene brightness.

We used this concept of exposure adjustment based on absolute scene luminance to achieve a good exposure. It builds the artistic decisions made while photographing scenes into exposure rules for the camera to follow. This results in scenes that display darker or lighter than middle gray depending on lighting conditions. This is what we'll be discussing next.

7.2 Exposure control curve

We define an exposure control curve (Figure 9) using three points, the dark, mid and light points. Each point maps some average scene brightness (in EV) to display brightness (in EV), and the curve as a

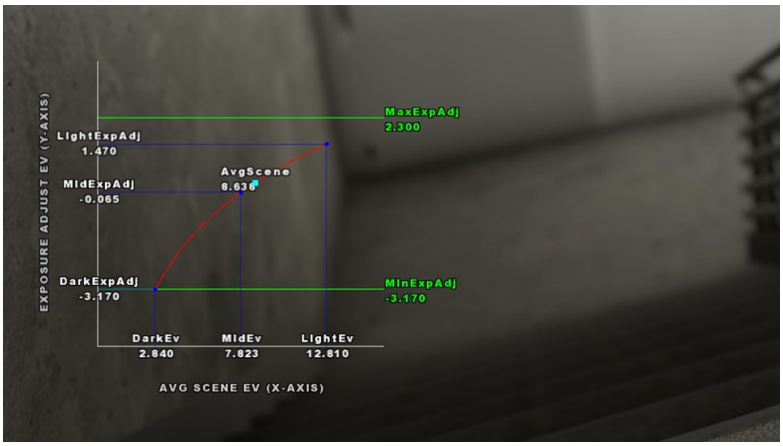

Figure 9: Three points define an exposure control curve: dark, mid and light. The curve maps average scene luminance (in EV) to exposure adjustment (also in EV), which changes display brightness. Exposure adjustment is clamped by the two horizontal green lines.

whole acts as a function to convert between the two. It's important that the curve is monotonically increasing, since we never expect the display to get brighter as the scene luminance gets darker.

Using EVs to express brightness on both axes of this curve is intuitive since we already deal with EVs in all stages of lighting. Originally, we described this curve as a function of linear scene luminance to photographic image key[3], but that proved to be cumbersome and hard to grasp.

It helps to illustrate two possible extremes for the exposure curve. If you imagine a horizontal line for your curve, then you would not have any exposure adjustment and the resulting images would maintain the same average brightness. If you imagine a line with slope 1, then there's absolutely no adaptation. In this case, your exposure is fixed and won't change in response to changes in lighting.

This exposure curve is a simple idea but it was key to achieving a good exposure for us. The curve simulates, in a simplistic, parameterizable way, the decision making process that occurs when a photographer sets exposure. We do this so the game can react to changes in lighting without explicit scripting.

7.3 Ideal Exposure

One of the common problems you run into when determining exposure is metering on a very dark or very light surface with no exposure adjustment. This has the effect of making that surface display at middle gray, making the entire image look under- or overexposed. For an ideal exposure, we'd like to determine exposure based on lighting luminance, unmodulated by the effects of material properties like diffuse albedo. For example, we'd render a lighting buffer containing only the effects of lights[4] in the scene, and we would meter this lighting buffer, rather than the scene buffer. In this way, it doesn't matter if a wall is painted black or white; exposure determined by lighting (and not diffuse albedo modulated by lighting) will render that wall appropriately.

An interesting property of defining an exposure control curve is you get some of this same effect because of the constraint that the curve is always increasing. Since white walls reflect more light, exposure adjustment will create a brighter image. While this isn't a direct replacement for exposure using a lighting buffer, we found that we could avoid the need to render a separate buffer.

[3]Where a key of 0.18 was our middle gray.

[4]Direct and indirect, diffuse and specular.

7.4 Adaptation

One of the effects you can achieve with varying exposure is the simulation of eye adaptation. When we transition from a bright environment to a dark one, our eyes adapt to compensate for the loss in brightness. At first we're not able to make out details in the darkness, but over time our eyes adjust until we can discern detail again. In the same way, we can have our in-game camera adjust exposure so that we don't suffer large changes in perceived brightness.

One of the common misconceptions concerning HDR is that its purpose is to provide this adaptation effect. Instead, we prefer to think of adaptation as part of the process of mapping HDR to a low-dynamic range device. In many cases you could argue that eye adaptation is counter-productive to an immersive experience, since the adaptation that needs to occur to output to a low-dynamic range screen is more extreme than what occurs in our visual system, which can sense a higher dynamic range.

This brings up some complicated questions: Is our goal to convey the sense of "being there"? If so, would simulation of perception through our eyes be more appropriate, given we don't fully understand how that translates to the screen, and given our expectations of reality on-screen are informed by our experiences with television and movies? We believe this is a rich direction of future exploration.

7.5 Consistency in brightness relationships

The reason we render in HDR is because this reflects the large dynamic range we can have in the real-world and this allows us to use a *consistent tonemapper* to translate those values to the display so that *brightness relationships are maintained*. Maintaining these brightness relationships is critically important.

Because we can maintain these relationships with HDR, it's easier to achieve photorealism. We simulate how light bounces around a scene, with real-world values, then we simulate the tonemapping that occurs in a camera and we end up with a *plausible* image.

Before HDR, when we dealt with rendering directly to the screen, many material and lighting settings were effectively used as tonemapping controls. Specifically, being able to tune your light attenuation function was required because we rendered straight to the display. Previously, we even had control over Fresnel exponent in the material so that we could tune materials to have a plausible specular response given LDR lights.

The problem with many individual controls is that it's difficult to be consistent, especially when we have many parameters which are tuned to taste by many different people. The other issue is these materials ended up being tuned for specific conditions and wouldn't necessarily look good in others.

Once again, what we can achieve with HDR rendering is tonemapping consistency.

8 HDR photography

We make use of HDR photographs in several parts of our pipeline and photographing a noise-free, linear HDR image is vital for accurate results.

8.1 Bracketing and Image Generation

We use a software package called PTGui for both HDR sky capture compositing and HDR single image generation. The images provided as input to PTGui for HDR generation need to differ only in brightness, so changes in depth of field are not allowed. Because of this, we bracket our shots (Figure 10) using changes in shutter speed rather than aperture, since changing aperture changes depth of field.

PTGui provides a *True HDR* option that outputs linearly correct values (Section 8.3) from input photographs and this is what we use for generating all our HDR images. We've also verified that an

Figure 10: Nine bracketed shots from -4 EV to +4 EV.

alternative process involving `dcraw` [Cof] and `hdr_create` [Cot] also generated linear images[5].

8.2 RAW vs. JPEG

We use camera RAW files for all our captures since cameras perform transformations on the colors before outputting JPEG. Those transformations are what give each camera manufacturer a different color rendition. Since these transformations are not documented, we rely on RAW files instead.

As an example of a RAW file pipeline, RAW sensor values are transformed in `dcraw` to CIE XYZ using a transformation matrix provided by Adobe DNG. Then these XYZ triplets are converted to our target color space, e.g. sRGB. How Adobe characterizes different sensors is unknown but we're *assuming* better consistency than relying on camera JPEG.

8.3 Verifying Linearity

We verified that our workflow for generating HDR photographs was outputting linear values using a Stouffer step wedge. The step wedge is a thin-film with squares of differing, graduated densities such that each square lets in half a stop more or less light than its neighbor and the entire sequence of squares spans a range of 10 stops.

By capturing an HDR photograph of the step wedge against a constant color light source we can determine if the pipeline used to generate HDR photographs is actually generating linear values.

In our case, we backlit the step wedge with an iPad set to display pure white. We also masked out any extraneous light from the iPad to reduce lens flare. The only light source in the photographed scene should be light coming through the graduated steps. We used the Perplexicon (Section 9.5) to shut out all external light sources.

For greater precision, a photograph of the masked screen of the iPad, without the step wedge, can be used to normalize the image to account for any spatial variation in the lighting.

9 Diffuse Albedo Texture Capture

Capturing diffuse albedo textures was helpful in creating a library of base materials from which we could gain an intuition for proper albedo values for authoring physically based materials.

9.1 Spectralon

The key ingredient in this process is Spectralon [Wikh]. Spectralon is a teflon-based pressed powder that comes closest to being a pure Lambertian diffuse material that reflects 100% of all light[6]. If we

[5] We used `dcraw` to generate 16-bit linear images from RAW files and then processed these images using `hdr_create`.
[6] It actually reflects greater than 99% over the range 400 to 1500nm (see [Wikh]).

Figure 11: An example of diffuse capture of a concrete material. Notice the perfectly white Spectralon in the top left corner.

take an HDR photograph of the Spectralon alongside the material to be measured, we can derive the diffuse albedo of that material.

9.2 Process

The process to capture diffuse reflectance is very similar to the one outlined by Hable [Hab10].

1. We put a linear polarizing filter in front of the camera lens and a second linear polarizing filter in front of a modeling light or a flash such that the two filters are oriented perpendicular to each other, i.e. cross polarized.

2. We place Spectralon close to and parallel with the material we are capturing and take bracketed shots of the setup[7]. Typically, we'll take nine photographs, from -4EV to +4EV in 1EV increments.

3. We convert the bracketed shots to a linear HDR image. We found that many HDR packages do not produce an HDR image in which the pixel values are linear. PTGui is an example of a package which does generate a linear HDR image. At this point, because of the cross polarization, the image is one of surface diffuse response.

4. We open the file in Photoshop and normalize the image by color picking the Spectralon, filling a new layer with that color and setting that layer to "Divide". This sets the Spectralon to 1 in the image. All other color values are relative to this so we can consider them as diffuse albedo.

The light we use likely has an angular light falloff, being most intense in the center of the spot and falling off towards the edges of the cone. This unevenness in lighting can be accounted for, if necessary,

[7]It is not strictly necessary to capture bracketed shots for a high dynamic range image here. In theory, you could also take a well-exposed photograph and recover your linear values from the RAW data. We capture and process HDR to guarantee more consistent results.

Figure 12: Left: A photograph of carpet. Right: The diffuse albedo texture for this carpet. Notice the increase in saturation when compared to the photograph.

by taking a photograph of a white surface[8], using the same lighting environment and using this image to normalize the captured images.

We consider the result of this process to be diffuse albedo because we assume diffusely reflected light becomes unpolarized: the light scatters under the surface, is partially absorbed, becomes unpolarized and then refracts and exits the surface. This unpolarized light passes through the filter on the camera lens, whereas polarized light from specular reflection will be blocked by the filter. The interaction of light at the surface interface is actually more complicated than described above and some small fraction of diffusely scattered light on refraction and exit can become polarized [Wikb]. Also, we haven't considered multiple bounces of the light, which complicates matters further, nor the efficiency of the polarizer pair, i.e. the actual extinction resulting from the cross polarization. Given a light source such as a camera flash, we are also assuming a flat surface since surface microfacets not oriented perpendicular to the light rays will receive less light per differential area. We expect that these effects lead to small errors in our diffuse albedo but accept them.

9.3 What to expect from diffuse albedo

If you were to display diffuse albedo in sRGB, typically it would appear more *saturated* than the material appears in real-life. This makes sense since the addition of specular reflection and the tonemapping operation can both have desaturating effects on material appearance (Figure 12). In the case of specular reflection, we are adding (usually white) light. When tonemapping, each of the three color channels generally approaches 1 asymptotically.

The key idea here is that diffuse albedo doesn't look exactly like the material in real-life. If you were to try to paint diffuse albedo as it appears, the material will look wrong once rendered.

9.4 Challenges in painting diffuse albedo

Our initial idea was to tune materials by rendering the most neutral, desaturated image we'd ever want, without color grading. Color grading would be the stage where aesthetic decisions were made, such as an increase in contrast or saturation. To provide a neutral image, we used the Reinhard tonemapper.

We ran into an issue when we first started painting diffuse albedo for the Golden Gate Bridge. The Golden Gate Bridge uses a color called International Orange, which is very saturated. This is a very difficult color to paint correctly when viewed using Reinhard, which didn't meet our artists' expectations of the saturation perceived in real-life or on the screen.

[8]For example, a surface with a water-based barium sulfate coating, which has $> 97\%$ reflectance.

Figure 13: (left) The Perplexicon: a light-tight box made of black foamcore. (center) Baffles reflect and redirect light away from sample. (right) Side-view showing light path.

Rather than use color grading to adjust contrast and saturation, we moved to an S-curve [Day12] for tonemapping. We gently introduced a bit of toe to the curve and this became our new 'neutral' image. Rather than allow full control over all aspects of the tonemapping curve, for the next project we'll probably move to a preset system, with each of the preset curves representing 'film stock'.

This brings up an interesting question: How do you author something that is one step removed from its final look, i.e. how do you paint something that will undergo transformations before being displayed? Selan [Sel12] touches upon[9] some of the same challenges we encountered. One possible solution is a reverse tonemapper and 'de-lighter' tool, where artists could author the final look and the tool would perform the transformation into diffuse albedo.

9.5 The *Perplexicon*

When capturing diffuse albedo textures, all light should be linearly polarized, so we performed most of our captures either in a completely dark room or—for samples that were outdoor that couldn't be moved—at night, away from non-polarized light sources. Many times though, we couldn't move the sample. For example, we wanted to capture concrete and paint diffuse textures within our office stairwell. We weren't able to turn off the lights in the stairwell so we built a light-tight box out of black foamcore to shut out all extraneous light (Figure 13). Unfortunately, the light from the flash bounced off the interior surface of the box, causing it to become depolarized.

Since our diffuse capture method relies on pure polarized light hitting our sample, we thought about how to reduce the reflectance of the interior surface. Our first instinct was to coat the interior of the box with a low-reflectance material, such as black velvet or a dark black paint. Instead, we added baffles[10] to the box, which used the same foamcore material, such that reflected light would bounce many times before hitting the sample (Figure 13). Since the box was near black, several bounces were enough to reduce the amount of depolarized light so that it didn't measurably affect the diffuse capture. We believe this is a general rule of thumb: using geometry to redirect and bounce light multiple times off a moderately low-reflectance surface is more effective than relying on a single bounce from a very low-reflectance material.

10 Specular Gloss Capture

A glossmeter is a device for measuring the glossiness of materials. We use a glossmeter (Figure 14) that shines a light at a surface at 20, 60 and 85 degrees and measures the light in the reflected direction

[9] See the "Texture and Matte Painting" section of Selan's course notes.

[10] Baf·fle /ˈbafəl/ verb. totally bewilder or *perplex*. noun. a device used to prevent the spreading of sound or light in a particular direction.

Figure 14: Left: Glossmeter used to measure glossiness of materials. Right: The black glass used to calibrate the glossmeter.

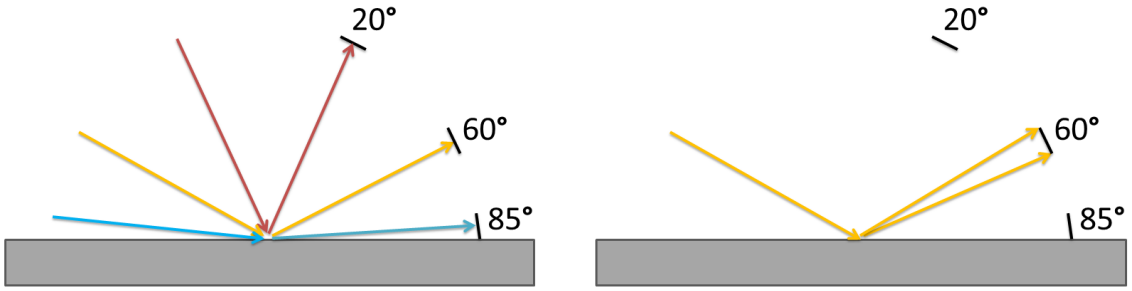

Figure 15: Left: Angles at which reflectance is measured. Right: ASTM D523 defines the aperture of the emitter and receiver. We treat the emitter as a point light source but integrate over the area of the receiver.

(Figure 15).

All measurement values are relative to a calibration standard: a reference black glass. The readings are in gloss units (GU). The reference black glass reflects 100 GUs at each measurement angle. If, for example, the material we are measuring returns 10 GUs at 20 degrees, this means the material reflects 10% of the light of the black glass at that angle.

We take the measured GU of a material and calculate in-game gloss values based on an in-game reference black glass. A significant assumption we make here is that our in-game black glass is a faithful representation of the real-world one.

Our in-game black glass is a standard GGX material with gloss = 1 (or, equivalently, roughness = 0) and index of refraction = 1.567 (provided by the manufacturer of the glossmeter). We calculate the reflectance of this theoretical black glass at 20, 60 and 85 degrees, and then treat these reflectances as 100 GUs. Then, calculating the gloss from measured GUs becomes an inverse rendering problem: we search for the in-game gloss that returns the same GU as our measured material.

The ASTM D523 standard [AST] dictates the placement of the emitter and receiver apertures that take the readings in the glossmeter [NET]. While we integrate over the aperture of the receiver to calculate reflected light, we treat the emitter as a point light source (Figure 15).

When measuring a material, we get three different GU readings and so calculate three different gloss values. The more those gloss values differ, the less well suited GGX is to that material. Real-world measurements of gloss at those three angles typically span a good range, which highlights the deficiency in parameterizing something as complex as roughness or smoothness one dimensionally. We believe a good future direction for research would be to parametrize glossiness multidimensionally, although artist authoring of other dimensions may prove to be difficult (Section 18).

There are many caveats to this technique of gloss measurement. The gloss values are a rough first approximation; they are useful for determining broad, relative gloss relationships, but there is still some

Figure 16: Examples of sky boxes in-game, lighting the environment. The placeholder models fit into the scene.

choice to be made on which angle to take your gloss measurement from. Alternatively, it's reasonable to choose a gloss value within the range spanned by the measurements.

The gloss meter that we use doesn't filter out diffuse reflectance and we ignored this rather than complicate the workflow with an added diffuse measurement. This results in an overestimation of gloss since diffuse reflectance is treated as "extra gloss". Normally we feel this overestimation is minor except in the case of very rough, lighter colored surfaces such as concrete, where the measured gloss will be obviously too high.

If a surface is too rough at larger scales, like stucco, then the gloss measurement can vary wildly depending on where we took the measurement since we don't measure normals and adjust accordingly, i.e. we assume a flat surface.

In spite of these caveats, we found the measurement of gloss useful for furthering our understanding of specular behavior and for gaining an intuition for plausible values.

11 Sky Capture

The sky is a large source of lighting for our scenes so getting realistic values in our skyboxes is really important in achieving photorealistic lighting. Initially, we experimented with software packages to generate our skies, but we were unable to get the software to output realistic linear values.

We ended up capturing all of our skies using a camera mounted on a spherical panorama head on a tripod. We bracketed our shots and converted and composited using PTGui. The sun values lay far outside the range of our bracketed shots so we painted it out. This works well with our system since we already treated the visual aspect of the sun separately as a sun sprite in-game. We also light the scene using a punctual sun light source so we don't want a sun in the sky contributing to lighting, i.e. double lighting.

12 Tuning materials using HDR photography

We have a feature in the game that allows us to render an HDR photograph before the tonemapping stage. By applying the same tonemapper, we can make a fair comparison between a rendered scene and an HDR photograph.

By recording EV readings of colorcheckers (or representative surfaces) within the HDR photograph while capturing it, we can scale the HDR photograph such that it renders correct luminances into the scene buffer. We are able to do this since we can spot meter the HDR photograph to measure EV. This makes matching the real-world luminances of surfaces a matter of tuning our in-game materials to match.

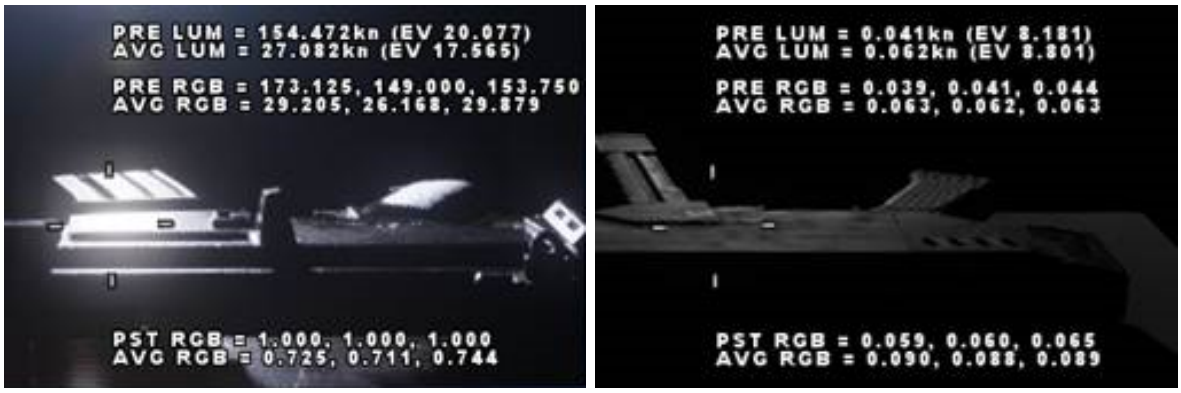

Figure 17: Left: HDR photograph of an airsoft rifle. Highlight measures 17.5EV. Right: Original material used on in-game gun. Highlight measures 8.8EV.

12.1 Gunmetal

Partway through development, most of our materials were looking pretty good but there were certain materials that we judged to be too diffuse, or *chalky*. To investigate, we decided to examine a gunmetal material which we believed suffered from this issue. We took some HDR photographs of a realistic looking airsoft rifle illuminated only by a single modeling light from different angles. Spectralon was placed next to the airsoft rifle and its EV was measured for calibration purposes. We brought these HDR photographs into the game, scaled their color values up to give an accurate luminance for the Spectralon, and began measuring the highlights from the light at different angles. Then we recreated the scene with the gun and the modeling light in-game and measured the in-game luminance values.

What we found, in the worst case, was almost a nine stop difference in the highlight luminance between our in-game weapon and the HDR photograph of the airsoft rifle (Figure 17). By tuning the glossiness of the weapon material, we were able to get our luminance values within one stop of our HDR photograph. At this point we had a good visual match since a one stop difference in a bright specular highlight is not really noticeable.

12.2 GGX

Originally, our shaders used Blinn-Phong, but during this process we also experimented with GGX and determined the gradual light falloff to be more visually pleasing and a better match for many of our materials (Figure 18). Our artists also reported they saw better material differentiation with varying gloss values. We switched over to GGX at this point, so these tests motivated a fairly large change to our shaders.

In order to avoid retuning all our existing gloss maps, we reparametrized α for normalized GGX by peak matching our original normalized Blinn-Phong (with exponent m), as suggested by Treyarch[11]:

$$\alpha = \sqrt{\frac{2}{m+2}} \tag{4}$$

Where, for our original Blinn-Phong

$$m = 2^{16gloss} \tag{5}$$

This parametrization has some nice properties: α is never 0, and glossiness is interpolated in the same manner as before, which to our eyes is perceptually uniform.

[11] Personal communication.

Figure 18: Example of GGX applied to the top part of the gun along with recalibrated gloss. Notice the appropriate specular falloff for gunmetal. For comparison, the bottom of the gun uses Blinn-Phong with original gloss.

Figure 19: Examples of a car in four different lighting environments.

13 Tuning materials using standard lighting environments

We tune our materials in a test level that includes several standard lighting environments. An artist is able to view their model or individual material (applied to a placeholder model) in each of these environments to assess its visual accuracy.

Most materials are tuned within our sunny day environment and are checked against the overcast, sunset and night environments (Figure 19).

One potential issue with our standard lighting environments is they are composed entirely of a skybox. Because of this, there is no view-dependent lighting from the background—such as specular reflection—to gauge the tuned material against. Without this frame of reference, there's the possibility that material artists could bias towards less glossiness. This is something we are looking to address as we work on our next project.

14 Retroreflective Diffuse

Measurements are used to validate computed values, but they can also be used to motivate more research and changes to rendering. We made an unexpected observation when measuring stucco concrete around our office building.

The strange thing we noticed was that the wall would appear brightest when the light was at a grazing angle, but your back was to the light (left-most image in Figure 20). This goes against what we would typically expect: a large grazing angle specular response in the right-most image, which would

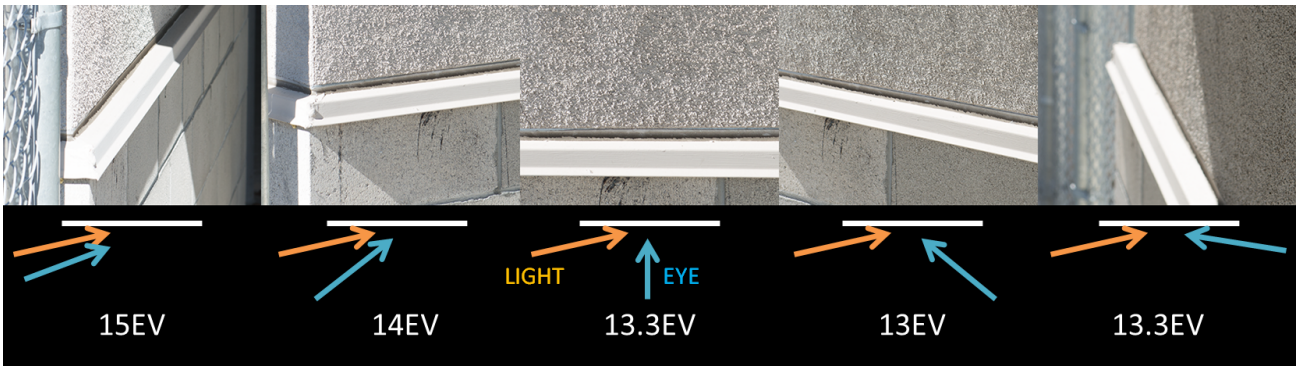

Figure 20: Unexpected measurements made at different angles to a bumpy wall. At far left, the wall is brightest (15EV) when the light is behind the viewer.

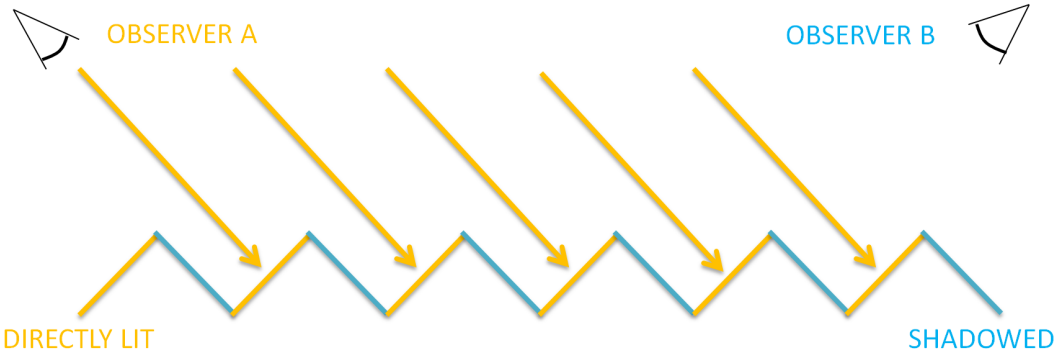

Figure 21: Extreme example of lighting a bumpy surface. Observer A sees the surface as lit while observer B sees the surface as shadowed.

result in the highest EV reading.

Lambertian diffuse, which is constant and view-independent, is a poor model for this rough surface. The pictures in Figure 20 show off the bumpiness of the wall, but those bumps are not quite at the resolution of our normal maps. Even if they were, we'd still have to handle reducing resolution for mipmapping. These bumps also create micro-occlusions that are not modeled by Lambertian diffuse, and change the normal distribution that is visible to the eye (Figure 21).

Because of this observation, we explored using the retroreflective diffuse model in [Bur12]. Retroreflective direct lighting was straightforward to implement but at the time we didn't have a solution for handling retroreflective indirect lighting, so neither made it into the shipping game.

Since rough concrete surfaces constitute a large portion of screen real estate in a typical scene, this is something that we're considering revisiting for the next project.

15 Offline Rendering

While our ground truth is the real world, it's useful to perform intermediate validation by rendering scenes offline using a path-tracer such as Arnold. We strive to match our offline render with the real-world and then our real-time render with the offline one. In our test of the Cornell box we compared our measurements of the wall luminance in our Arnold render against the original measured values [Cor] and they matched up well. Then we constructed the same Cornell box in-game and ensured our values matched both the real-world measurements and the rendered offline values (Figure 22).

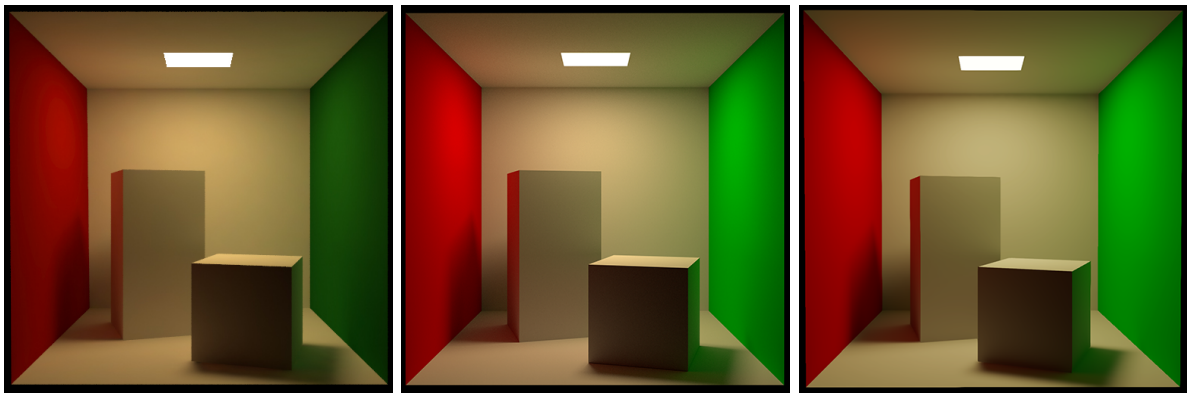

Figure 22: From left to right: original synthetic image from data, offline render using Arnold, real-time in-game render

16 Color Space

16.1 Spectral Power Distribution

Perceivable real-world colors are comprised of a spectral power distribution in the visible wavelengths. We approximate colors using triplets, or tristimulus values, in our real-time rendering. Typical multi-spectral rendering approximates the distribution as a series of 10nm wavelength buckets, so we can think of real-world colors measured at our digital sensors going through a transformation from a high-dimensional space into three dimensions. This transformation is inherently lossy but is similar to the transformation that occurs in the human visual system. The lossy nature results in metamerism where different spectral power distributions appear as the same perceived color.

The three colors typically used as a basis for tristimulus values are red, green and blue; these are similar to the spectral sensitivity peaks of the three kinds of cone cells in our eyes. It's important to recognize there are different shades of red, green and blue which form the bases of our color spaces. Many color spaces, such as sRGB [Wiki], have their bases defined by colors in the CIE XYZ color space [Wikc]. Because of this, sRGB and CIE XYZ inhabit the same three dimensional space and you can transform between them using linear transformations.

If the red, green and blue filters in a Bayer pattern digital sensor cannot be represented in CIE XYZ, then the step to transform RAW sensor values into sRGB will also be lossy.

Chromatic adaptation, or white balance, when done with triplets, is also an approximate, lossy operation since we are adapting to illuminants with arbitrarily complex spectral power distributions.

16.2 Capture Color Space Consistency

When performing diffuse albedo texture capture and sky capture, it is important to maintain a consistency in color space and white point between the two. When dealing with lighting values as RGB triplets, the primary chromaticities used should match between your different captures.

To use an extreme example, let's assume your skies were captured in a cyan-magenta-yellow color space and your diffuse albedo textures were captures in a red-green-blue one. The sky is used to light the scene, but it wouldn't make sense to take those CMY triplets and multiply them by RGB in the shader. In the same way, triplets with different RGB primaries shouldn't be multiplied by each other. Standard sRGB (with a white point of D65) has different primaries from sRGB chromatically adapted to a white point of D50, so are incompatible. For this reason, we standardize on all captures with sRGB/D50 since this gives us a natural sky with no chromatic adaptation. We set custom white balance in-camera to 5000K so that we have white balanced JPEGs for reference.

Figure 23: This is an early whitebox, with minimal art, after all the elements came together. The details may not be there, but the broad impression of the scene feels right.

16.3 Chromatic Adaptation

Chromatic adaptation, or white balance, was not something we dealt with on Advanced Warfare. Most light colors were tuned to taste and we didn't provide a white balancing shader. This is a dimension of control that we are interested in exploring for our next project.

17 *Squintworthiness*, or putting it all together

Once all the elements come together, the resulting render should look correct (Figure 23). You can gauge an image to determine if the broad strokes are correct by taking several steps back from the display or by squinting your eyes. This has the effect of masking any deficiencies in detail that we tend to focus on. Then ask yourself, do you recognize this as a plausible image?

Early on in development, we were working on a level that is a faithful representation of our office building. At the time, it wasn't looking quite right: the lighting was off. We sat down and entered measured values and a captured sky and slowly but surely, with each iteration, the level started to look *right*. Once we were finished making changes, there was excitement in the air. A group of us were huddled around one monitor and there was unanimous acknowledgement: we had passed some threshold where the impression of the scene rang true. This was an inflection point in the project where the measurements we made paid off.

18 Future Directions

18.1 Specular tail control (Haze)

From our glossmeter measurements, we suspect that our specular response could benefit from an added dimension of control over the tail of the specular lobe. If gloss controls the narrowness of the specular peak, we could use *haze* to control tail falloff.

Löw et al. [Löw+12] suggest a simplified version of the ABC normal distribution function (NDF) first introduced in [CTL90] for tail control. A is a normalization term, B controls the sharpness of the specular peak—what we normally control with gloss—and C controls the tail falloff of the lobe:

$$D_{ABC} = \frac{A}{(1 + B(1 - \mathbf{n} \cdot \mathbf{h}))^C} \tag{6}$$

Burley [Bur12] introduced[12] a generalized Trowbridge-Reitz (GTR) NDF, which uses a varying exponent in the denominator of GGX to provide this control. We use the ABC convention above in the following

[12]See Section 5.4 of this publication.

adaptation of GTR:

$$D_{GTR} = \frac{A}{(1 + (B-1)(\mathbf{n} \cdot \mathbf{h})^2)^C} \tag{7}$$

These two specular models are interesting but are difficult to use in our game engine. We use pre-convolved cube maps for our indirect lighting contribution, with varying gloss (B) represented in the different MIP levels of the cube map. Providing an extra dimension of tail control would require some way of looking up—for example, through a cube map array—or calculating the correct indirect contribution given the haze parameter (C).

Fortunately, an interpolation between two normalized distributions yields a normalized distribution[13]:

$$t \int_\Omega D_1(\mathbf{h})(\mathbf{n} \cdot \mathbf{h}) d\mathbf{h} + (1-t) \int_\Omega D_2(\mathbf{h})(\mathbf{n} \cdot \mathbf{h}) d\mathbf{h} = 1. \tag{8}$$

We can interpolate between Berry[14] [Ber23] and GGX—where t is our haze setting—and still have a normalized distribution. The advantage of this new distribution is we can sample from two cube maps, one pre-convolved with Berry and the other pre-convolved with GGX, and interpolate between the samples to get our indirect contribution. This is an interesting direction that we may explore for our next project.

18.2 Material acquisition

Although we are able to get rough estimates of gloss using the glossmeter, these are only single readings. It would be great to be able to measure spatially varying gloss, like we do for diffuse albedo. We're exploring different techniques using digital cameras to capture gloss textures.

Beyond this, one of the challenges in providing specular tail control is the authoring of the haze parameter. A solution to measure haze is likely needed before we can comfortably add this control to our shaders.

19 Acknowledgments

Thanks to: Angelo Pesce, Jorge Jiménez, and Peter-Pike Sloan for many insightful discussions. Dave Blizard for putting the theory of measurement to practice and constantly pushing for better quality. Dimitar Lazarov for help and advice in our transition to physically based materials. Michał Iwanicki for coming up with a solution for retroreflective indirect diffuse lighting. Tommy Cinquegrano for creating the Cornell box renderings. Stephen Hill and Stephen McCauley for organizing this course, editing these notes, fantastic feedback and great insights.

19.1 Photo Credits

Figure 7: Starlit photograph courtesy of Eric Gofreed. Moonlit photograph courtesy of Florian Senand. Sun photograph is public domain, from NASA Solar Dynamics Observatory.

19.2 Contact

danny.chan@activision.com

[13] An insightful observation by Jorge Jiménez.

[14] This distribution is equivalent to GTR with an exponent of 1.

References

[AST] ASTM International. *ASTM D523-14, Standard Test Method for Specular Gloss*. URL: http://www.astm.org/Standards/D523.htm.

[Ber23] E. M. Berry. "Diffuse Reflection of Light from a Matt Surface". In: *J. Opt. Soc. Am.* 7.8 (Aug. 1923), pp. 627–633. URL: http://www.osapublishing.org/abstract.cfm?URI=josa-7-8-627.

[Bur12] B. Burley. "Physically Based Shading at Disney". In: *Practical Physically Based Shading in Film and Game Production, ACM SIGGRAPH 2012 Courses*. SIGGRAPH '12. Los Angeles, California: ACM, 2012, 10:1–10:7. ISBN: 978-1-4503-1678-1. URL: http://selfshadow.com/publications/s2012-shading-course/.

[Cof] D. Coffin. *dcraw*. URL: http://www.cybercom.net/~dcoffin/dcraw/.

[Cor] Cornell University Program of Computer Graphics. *Cornell Box*. URL: http://www.graphics.cornell.edu/online/box/.

[Cot] A. Cotter. *hdr_create*. URL: http://ttic.uchicago.edu/~cotter/projects/hdr_tools/.

[CTL90] E. L. Church, P. Z. Takacs, and T. A. Leonard. "The Prediction Of BRDFs From Surface Profile Measurements". In: vol. 1165. 1990, pp. 136–150. URL: http://dx.doi.org/10.1117/12.962842.

[Day12] M. Day. *An efficient and user-friendly tone mapping operator*. 2012. URL: http://www.insomniacgames.com/mike-day-an-efficient-and-user-friendly-tone-mapping-operator/.

[Hab10] J. Hable. *How To Split Specular And Diffuse In Real Images*. 2010. URL: http://filmicgames.com/archives/233.

[Löw+12] J. Löw, J. Kronander, A. Ynnerman, and J. Unger. "BRDF Models for Accurate and Efficient Rendering of Glossy Surfaces". In: *ACM Trans. Graph.* 31.1 (Feb. 2012), 9:1–9:14. ISSN: 0730-0301. URL: http://vcl.itn.liu.se/publications/2012/LKYU12/.

[NET] M. E. Nadal, E. A. Early, and E. A. Thompson. "Specular Gloss". In: *NIST Special Publication SP250-70* (). URL: http://www.nist.gov/calibrations/upload/sp250-70.pdf.

[Sch94] C. Schlick. "An Inexpensive BRDF Model for Physically-based Rendering". In: *Computer Graphics Forum* 13.3 (Aug. 1994), pp. 233–246.

[Sel12] J. Selan. "Cinematic Color: From Your Monitor to the Big Screen". In: *ACM SIGGRAPH 2012 Courses*. SIGGRAPH '12. Los Angeles, California: ACM, 2012, 9:1–9:54. ISBN: 978-1-4503-1678-1. URL: http://cinematiccolor.com.

[Wika] Wikipedia. *Aperture*. URL: http://en.wikipedia.org/wiki/Aperture.

[Wikb] Wikipedia. *Brewster's angle*. URL: http://en.wikipedia.org/wiki/Brewster's_angle.

[Wikc] Wikipedia. *CIE 1931 color space*. URL: http://en.wikipedia.org/wiki/CIE_1931_color_space.

[Wikd] Wikipedia. *Exposure value*. URL: http://en.wikipedia.org/wiki/Exposure_value.

[Wike] Wikipedia. *Film speed: the ISO 12232:2006 standard*. URL: http://en.wikipedia.org/wiki/Film_speed#The_ISO_12232:2006_standard.

[Wikf] Wikipedia. *Fresnel equations*. URL: http://en.wikipedia.org/wiki/Fresnel_equations.

[Wikg] Wikipedia. *Light meter*. URL: http://en.wikipedia.org/wiki/Light_meter#Calibration_constants.

[Wikh] Wikipedia. *Spectralon*. URL: http://en.wikipedia.org/wiki/Spectralon.

[Wiki] Wikipedia. *SRGB*. URL: http://en.wikipedia.org/wiki/SRGB.