AV4266 - Stunning Materials and Finishes: From Good to Great

Speaker: Pierre-Felix Breton, Autodesk Media & Entertainment

Class Description

This class will touch on various topics around materials and finishes that are defined for high-quality visualizations and renderings.

You will learn tips and techniques to define materials in Autodesk® 3ds Max® software by using accurate acquisition techniques while dealing with color calibration issues.

Also, we will demonstrate the use of procedural textures to introduce subtle, but important variations that bring images to life. Learn how to make wood look like wood, paint look like paint, and so on.

No more cartoon looks!

About the Speaker

Pierre-Felix Breton is a software designer who specializes in the field of physically-based lighting simulation and rendering. Currently employed by Autodesk Media & Entertainment, he participates in the creation of products such as Autodesk® 3ds Max®.

He also contributes to the design and development of materials and shader libraries included in Autodesk products where color consistency and physical accuracy is critical.

As his professional background includes electrical engineering, computer programming, and theatrical lighting, Pierre-Felix consults regularly on various architectural lighting design projects as a designer, technical coordinator, and simulation specialist.

Modeled in Revit, rendered in 3ds Max © Pierre-Félix Breton 2012

Contents

Color Management Basics

The first step in the definition of great materials and finishes for rendering starts with a proper strategy around clor management.

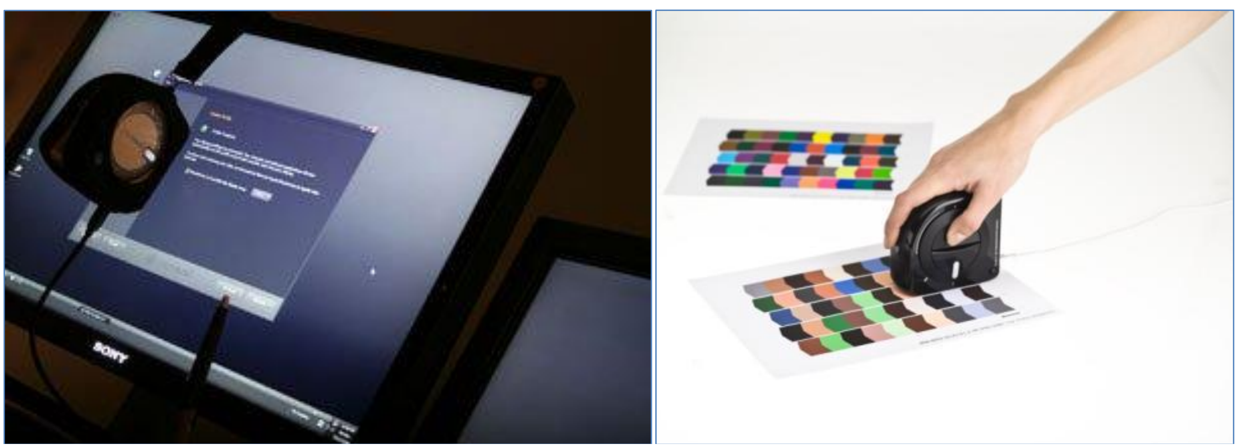
This section provides an overview of the techniques and limitations around color management and calibration. You will learn how to deal with the problem of capturing textures properly, load them in Autodesk® 3ds Max® appropriately and render out images with the correct color settings.

What tools are available to you?

Color management is there to insure that the colors you display on screen or on paper are consistent and coherent. A critical requirement is obviously associated with color measurement and control.

A (portable) spectrophotometer can measure the response curve of your computer screens and printers. This information is then looped back to your image editing software to obtain something closely matching an “idealized goal”.

One example of such device is the ColorMunki manufactured by X-Rite.



A portable spectrophotometer can define the response curve of your computer screens and printers.

Color management workflow example with Adobe Photoshop

Step 1: Profile your monitor

A color profile will be generated by your calibration device for your monitor. In our case, a ColorMunki is used to create a color profile that is automatically assigned in the display preferences of Windows via an ICC profile.

This process happens automatically so I will save you the details of where, how and what ☺…

If you do not have a spectrophotometer handy, you can minimally rely on built-in calibration wizards that come with your graphic card driver

For example my nVidia driver has a brightness, contrast and gamma adjustment wizard that does a good job.

Step 2: Profile your printer

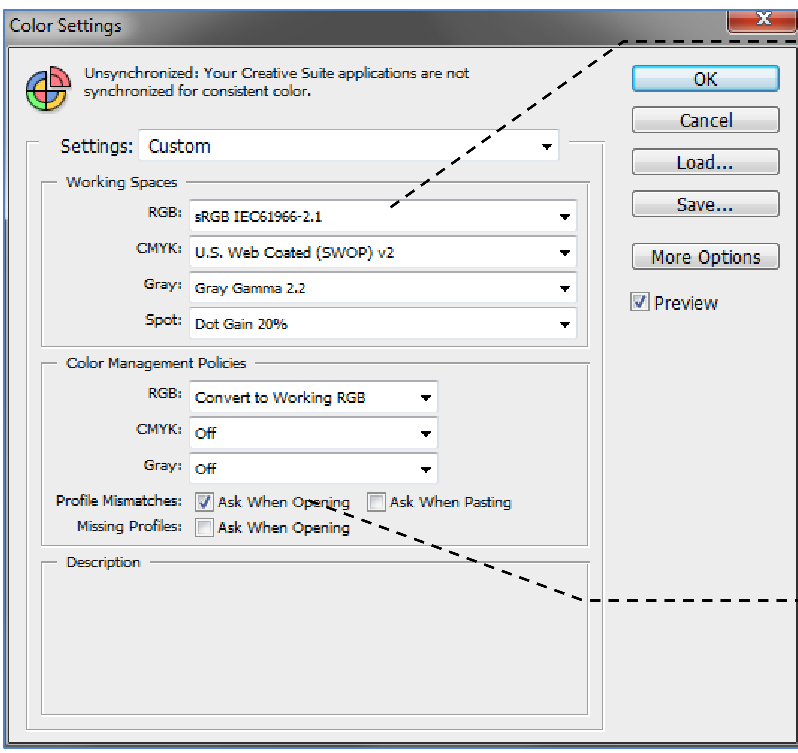
Similarly, the ColorMunki runs you through a wizard that prints a few sample swatches on a color printer. Those swatches are measured back and a color profile is generated. This profile is loaded/installed in Photoshop automatically as well.

Note: Repeat the process for <u>each type of paper you might use</u> in your printer as each profile takes into account the whiteness of the paper that was used during the calibration process.

Step 3: Enable color management policies in Photoshop

You are now ready to benefit from color management so you must instruct Photoshop to work with color management. This is available under Edit | Color Settings…

At this point, you also need to make a decision about the working space color profile. To remain as compatible as possible with 3ds Max, I recommend you to use to the sRGB IEC61966-2.1 profile as illustrated below.



If you are doing mainly work with 3ds max, or print images with services available in large stores (Wallmart, Cosco etc), I recommend you to base your work on the sRGB profile. This is the lowest common denominator that is supported everywhere.

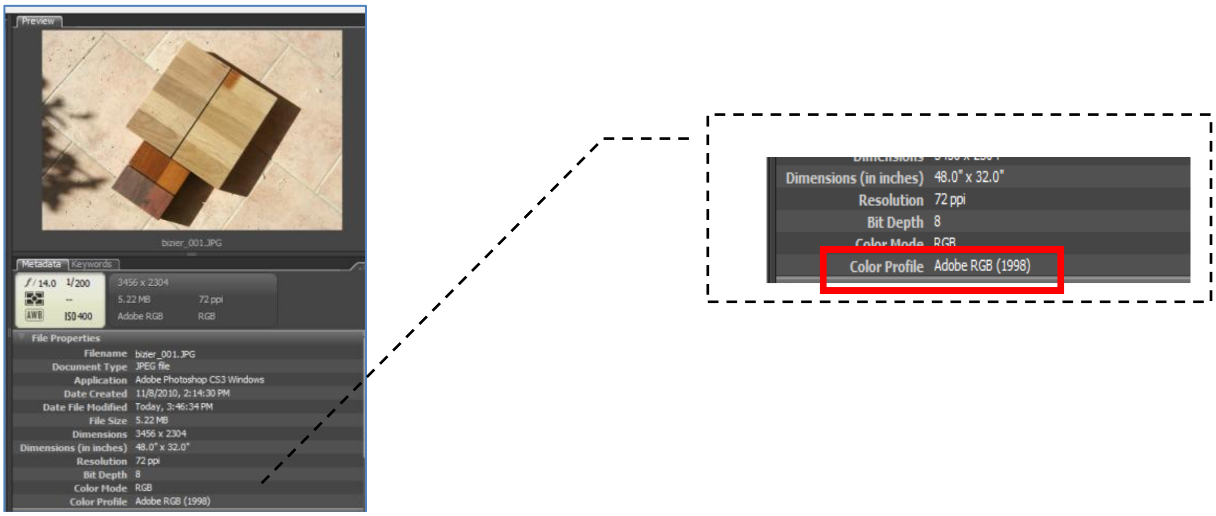
Gotcha!:

One common mistake to do here is to select the monitor profile that may have been generated by the monitor calibration device (Ex: Display01.icm created by a ColorMunki). Don't do that!

Turn on the "Ask when opening" option. You will be able to assign your default profile on file open if the incoming file profile mismatches yours.

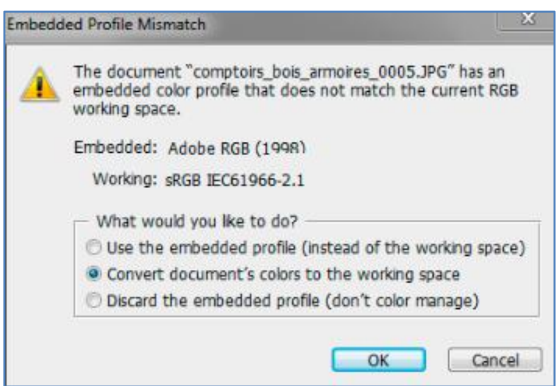
Step 4: Load an image in Photoshop

Most images created with digital cameras or scanners have a color profile embedded into them as Meta information. This color profile can be inspected in Adobe Bridge as illustrated here:



A color profile is usually associated with an image.

If you open an image that has a different profile than your working space (Adobe RGB vs sRGB in this current example), you may see this dialog:



Photoshop detected that the incoming image had a different profile than the workspace.

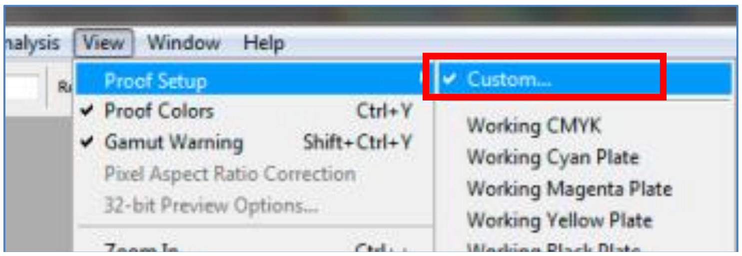
Knowing what each profile does is not really a concern to you in the moment. It is data that is used by Photoshop to convert pixel colors into something that closely matches the color gamut (range) of your display.

My recommendation is to convert it to the working space (because we established that 3ds Max is closely compatible to sRGB in a previous step).

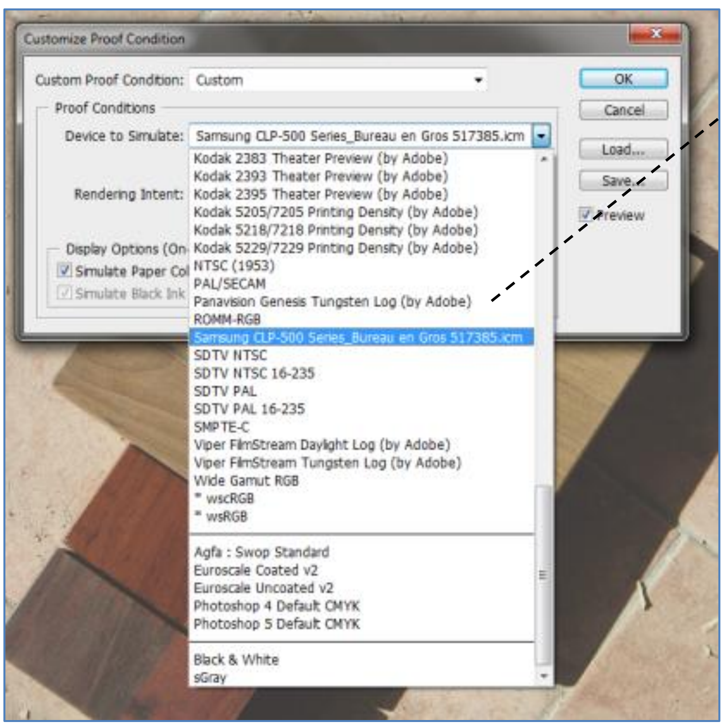
Step 5: Printing from Photoshop

Printing an image you see on screen requires what we call “Proofing” colors. The main reason is that the perception of color on paper is different than the screen: one emits light; the other absorbs light so they cannot be compared by using the same color profiles.

Fortunately, Photoshop has a built-in tool that can emulate colors that can be reproduced by your printer. This is available under View | Proof colors..



Configure the Proof Colors to match your target output (printer + paper profile).



This is where you use the color profile defined by the Printer Calibration step in Step 2.

You may have multiple types of papers. Remember to select the one that matches your target printer / paper combination.

Once you look at colors on screen under the “Proofing Colors” mode, you can go ahead and adjust the image to look like you want it to look.

If the image looks a bit faint, do not worry, this is normal: the “Proof Colors” mode try to emulate printed paper.

Since paper does not emit light like your screen, the result will necessarily be faint as well. This is where you need to fine tune the brightness and contrast of the image to give it a bit more punch.

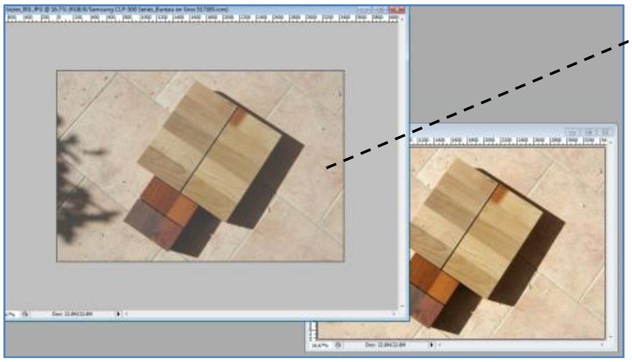
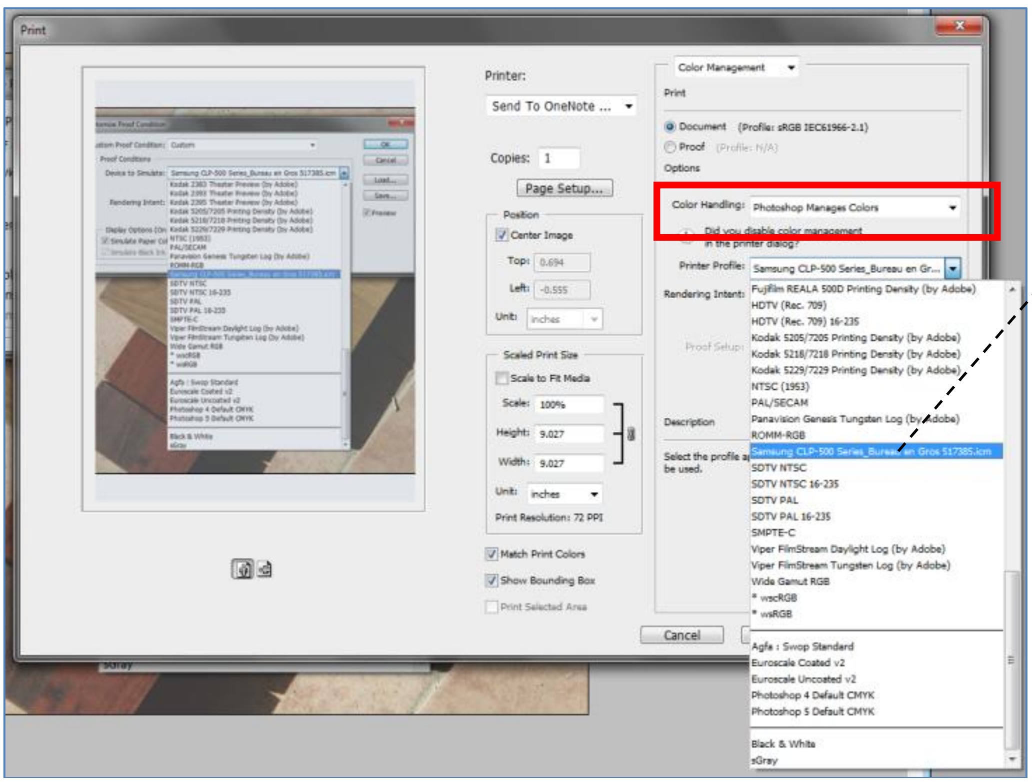


Image seen under "Proof Colors" mode, using a target output matching my printer profile.

If your intention is to print the image, adjusting final contrast settings should be done in that mode.

Images viewed under proofing vs. under normal conditions.

Once you are ready to print, instruct Photoshop to manage the colors, by using the same printer profile as the one used to proof the colors in the previous step:



Pick the same profile as the one you picked under the "Proof Colors" mode.

Photoshop Print Configuration Dialog with Color Management enabled.

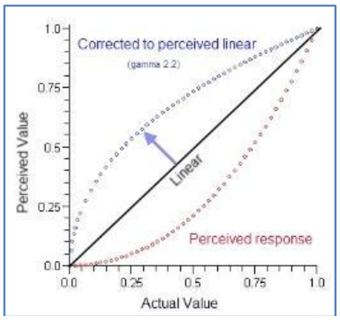
The resulting print output will be as close as possible to the image you saw on screen. Trying it is adopting it ☺

The importance of Gamma correction in rendering

Display encoding vs. Data encoding

Introduction to Gamma correction

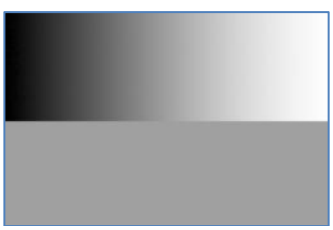
Display systems have their own inherent characteristics that are not perceptually uniform. Typically, images displayed “as-is” appear incorrect to the human eye. To counteract this, an adjustment is introduced. This adjustment is known as “*Gamma Correction*”.



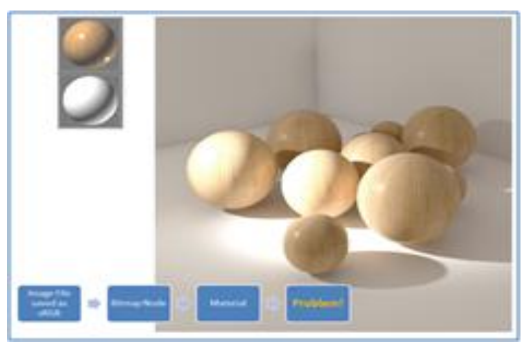
Gamma correction graph

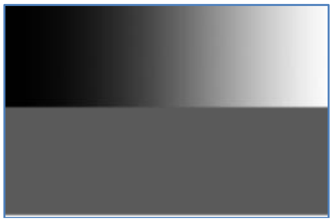
But here is the dilemma: images must be gamma corrected for display purposes (so they *look good*), but this creates a situation where physically incorrect data can be feed to 3D rendering engines. Typically, this problem occurs with diffuse colors for materials.

http://en.wikipedia.org/wiki/Gamma_correction

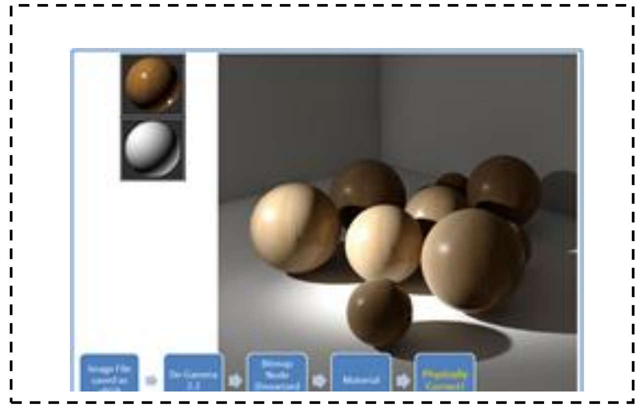


Gamma Corrected 50% grey:
Appear correct on display,
*feed **incorrect** (too bright) reflectance **data** to 3D renderers*





Linear Space 50% grey:
Appears too dark on display,
*feed **correct** (not too bright) reflectance **data** to 3D renderers*



Therefore, 3D rendering engines must "*De-Gamma*" the image textures prior to perform any light calculations. This *"De-Gamma*" correction will bring back the image textures into what we call "**LINEAR SPACE"**.

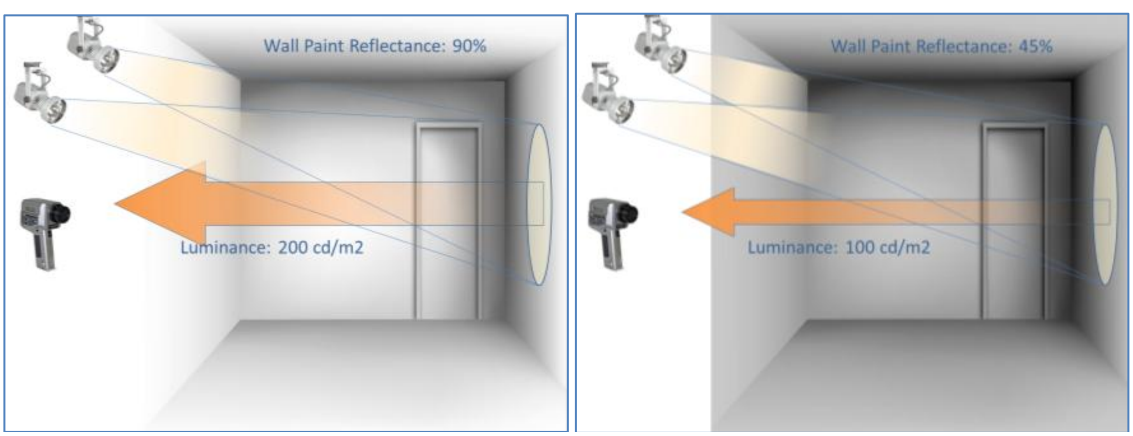
In summary, images used as <u>data to perform calculation</u> – or used to feed a numerical value to the renderer (diffuse textures, bump maps, normal maps) must be in *linear space* from the point of view of the 3D renderer while images used for display must be gamma corrected to look good (i.e. *non-linear space*).

Why is that? Because light is "linear"

When we use texture maps on materials, color management and Gamma correction becomes a core issue. The source of the problem lays into the fact that texture maps are stored and saved to look good on screen. However, this is not representative of the way internal math of render engine works.

One aspect that is critical to understand in 3D rendering is how light is interacting with material reflectance. In the following example, we demonstrate that changing the reflectance of the wall from 90% to 45% (divide by two (2)) also divides by two (2) the amount of light that bounce back in the room.

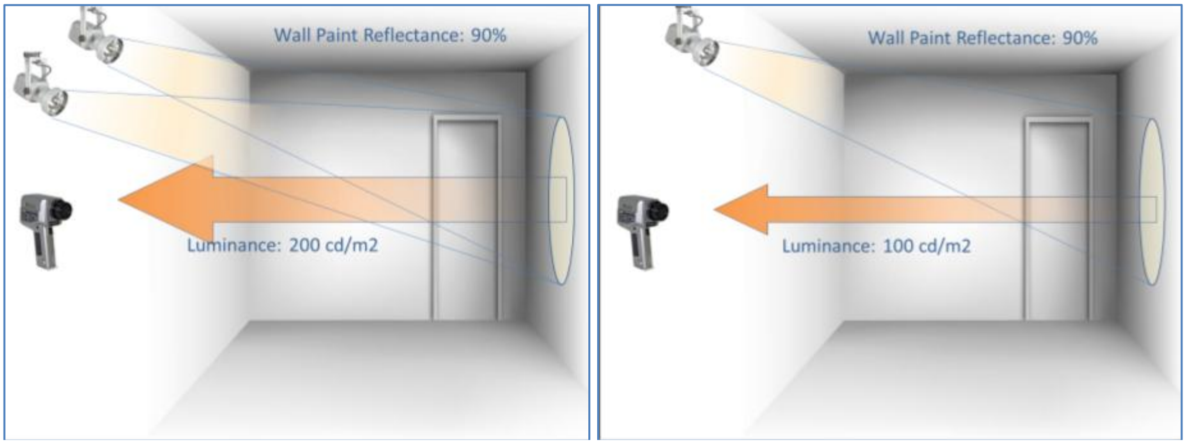
Light is **MULTIPLIED** with the reflectance (color) of the wall:



Making the color of a wall twice as dark will reduce the amount of light in the room by a factor of two.

Similarly, using a single light source instead of two will also produce the same effect: we get half of the light back in the scene as in the previous example.

Light can be **ADDED** or **SUBTRACTED**:



Using a single light source will reduce the amount of light in the room by a factor of two.

Here is simplified math behind the phenomena:

Wall Luminance (cd/m2) = Light Quantity (lux) * Material Reflectance (color) / PI

As you can see, this is simple math that only implies simple multiplications, divisions and additions. Light is calculated using linear algebra.

The more light sources you have, the more light you get. This is why we refer to it as **LINEAR SPACE**.

Guess what? 3D Rendering engines also work that way: they **ADD** *lights* together and **MULTIPLY** them with the *color* of the materials to obtain *luminance*.

3D Renderers calculate in **LINEAR SPACE**.

When should you apply Gamma correction on images?

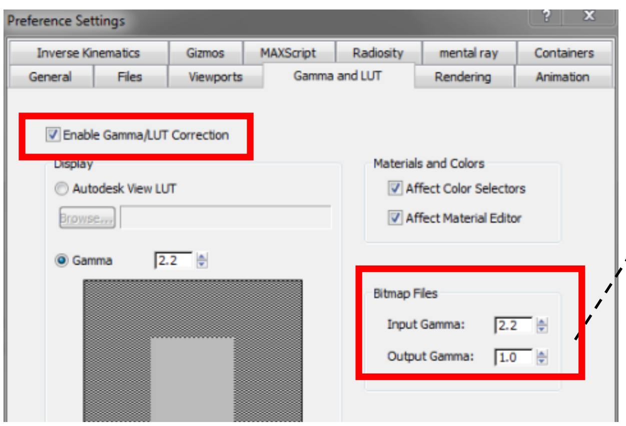
The general case

Since 3D engines calculate light just like reality, by using LINEAR equations, we must remove the Gamma encoding that was stored in images to compensate for the display.

In general, we are looking at the following data flow:

Data flow for a diffuse texture that "looks good" on screen passed to the 3D renderer in a physically correct way.

This is achieved in 3ds Max by enabling the Gamma and LUT preferences:



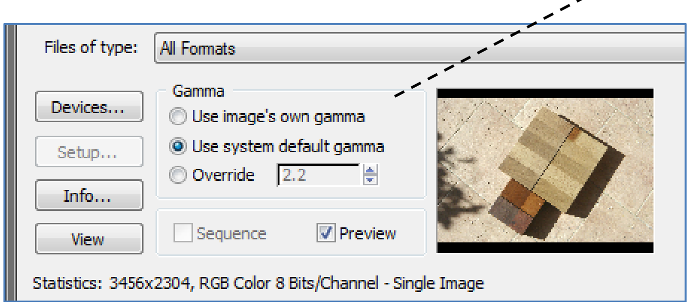
Input Gamma: when 3ds max *loads* an image from disk.

Leave at 2.2 for diffuse textures.

Output Gamma: when 3ds max *saves* an image to disk.

Leave at 1.0 if rendering out to HDR or EXR, 2.2 if rendering out to TIFF, JPG or PNG.

3ds Max: Customize | Preferences... | Gamma and LUT dialog



"System Default" will use the values from the Preferences dialog.

"Override" will let you override the Gamma values for a specific Bitmap. Typically used for HDR output (see below).

3ds Max: Image File Browse Dialog: this is where you can define a" per bitmap" Gamma correction value.

Gamma settings cheat sheet:

In doubt, you may want to refer to the table below to decide how to set the 3ds Max gamma parameters for your bitmaps:

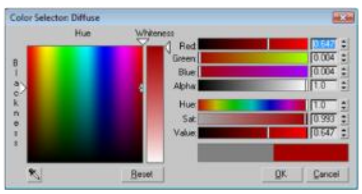
Application / Usage	UI Parameter	What will happen?
Diffuse, low dynamic range texture (wood map, carpet color etc.)	Use system Gamma	Map will be loaded with a Gamma 2.2, pulled from the preference dialog, which is what we want.
Bump, displacement, height maps, typically black and white, low dynamic range	Override : 1.0 on both input and render output (i.e. RTT)	It will bypass the Global settings and force the image to be loaded "as-is" without further correction.
EXR / HDR images	Override : 1.0 on both input and render output	We assume that HDR images are linear. By definition, 3ds Max does not perform any gamma correction on them. 3ds Max loads and save the data "as-is".

Materials

The 'problem' with color

While many graphical user interfaces allow users to specify RGB colors via color pickers, the resulting RGB value used by a simulation doesn't usually correspond to the color displayed on your computer screen.

As a result, picking a color based on the appearance of that color on your screen will lead to physically wrong outcomes.



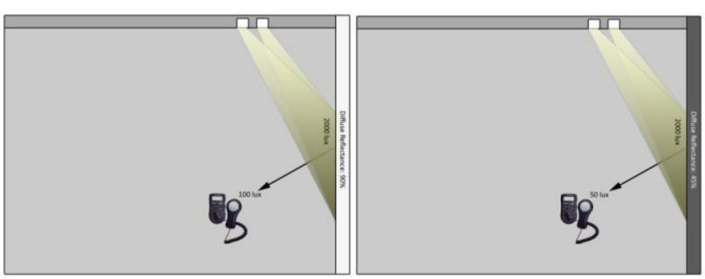
What is the right red to pick for this apple? It can be difficult to tell and trust what you see on screen.

It is really about "absolute reflectivity" not "perceived color"...

In the end, the apple will still reflect the same amount of light whether you look at it under bright sunlight or dim living room light. Your eyes won't perceive it the same way, but that doesn't mean that the apple's physical properties have changed.

No matter which type of material you use, the amount of light bounced back to the scene has a direct relationship with the material's color.

For example, a white wall reflects more light than a dark gray wall. A dark gray wall in sunlight reflects more light than a white wall in a dim room.



A white wall (90% reflectance) reflects twice as much light than a grey wall (45% reflectance). This is directly caused by the diffuse color of the materials.

This is the behaviour of diffuse reflectance. This is not to be confused with glossiness (for glossy or matte finishes).

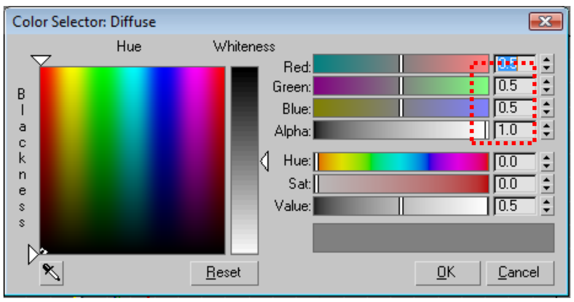
All indirect light bounces off surfaces in a diffuse manner in all directions (also known as “Lambertian”). In contrast, directed bouncing light, such as the pattern of a mirror in sunlight, is called a *caustic or specular reflection.*

Color is a measurement of how much light is reflected from an object

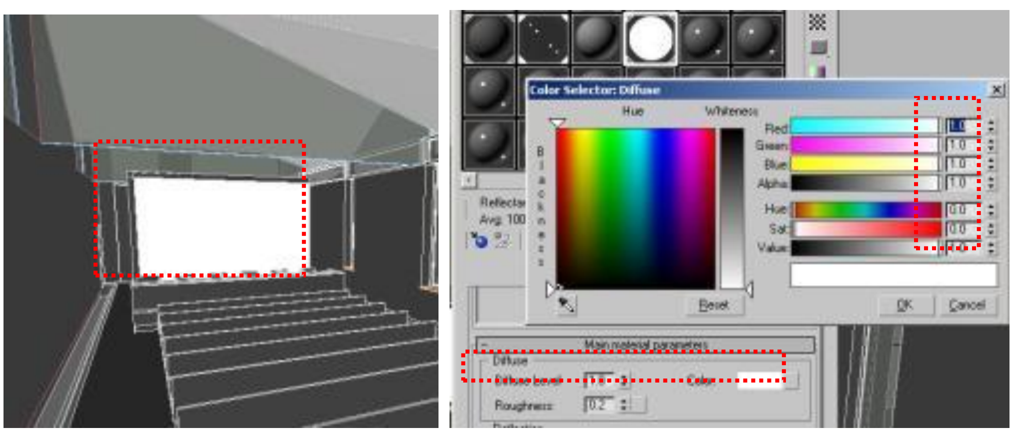
Typically, for physically based raytracers, the diffuse *color* of a material is in fact a diffuse *reflectance* value; where 0.5 means that the material reflects 50% of the incoming light for all R, G and B components’.

For example, a white wall will not reflect more than about 90% of the light it receives. Therefore, the RGB color should not be more than 0.90, 0.90, 0.90 (floating point colors) or 230, 230, 230 (integer colors).

To select the desired diffuse reflectance, select the tint (Hue) of your material and then rely on the “V” value, which usually closely matches the reflectance of your material.



A RGB value of 0.5, 0.5, 0.5 for the diffuse color of a material represents a diffuse reflectance of 50%. This is also mathematically equivalent to a Saturation of 0.0 and a Value of 0.5 in HSV space.



A material with a perfectly white diffuse color correspond (from a rendering point of view) to a diffuse reflectance of 100%. This is too reflective for a wall or a ceiling. Normally, white paint should be around RGB 0.90,0.90,0.90 (90% reflectance).

Estimating the diffuse color of materials

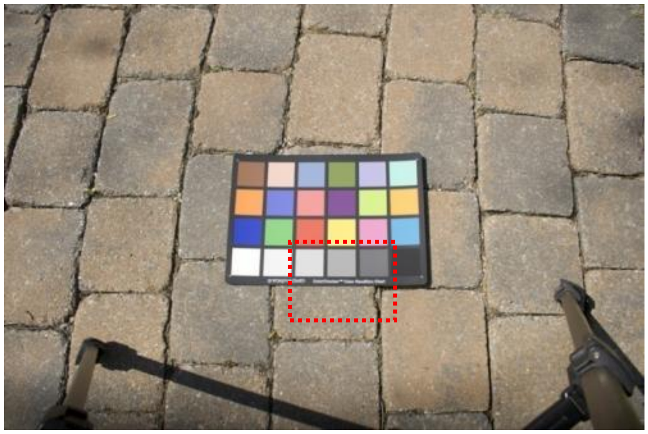
Using a reference chart

You can purchase a reference color chart that can help estimate the diffuse reflectance of materials by doing a quick eye ball comparison. A popular one, the GretagMacbeth ColorChecker™ is sold by X-Rite (http://www.x-rite.com) for roughly 80.00 USD. Those cards have calibrated colors and the reflectance of each swatch is published and known.

For example, the patch #22 (called "neutral 5") has a diffuse reflectance of 19%. This means that for a rendering engine, the correct RGB value would be R0.19, G0.19, and B0.19.

Patch name	Diffuse reflectance	Corresponding RGB color
white 9.5 (.05 D)	90.94	.9094
neutral 8 (.23 D)	58.50	.585
neutral 6.5 (.44 D)	35.71	.3571
neutral 5 (.70 D)	19.12	.1912
neutral 3.5 (1.05 D)	8.87	.0887
black 2 (1.5 D)	3.17	.0317

The industry standard GretagMacbeth ColorChecker™ has known reflectance values that can be used to compare against other finishes. Here, we are listing the reflectances of the lower row of swatches (white to black).



By eye balling, we deduct that the average reflectance of the pavement is roughly 20%: its brightness is close to the swatch #22 on the chart which has a knownreflectance of 20%.

Using a measurement device

I personally use a device called Color Munki manufactured by Xrite. This device allows me to acquire precisely the diffuse color of materials and finishes and store them into palettes that I can then import in 3ds max.

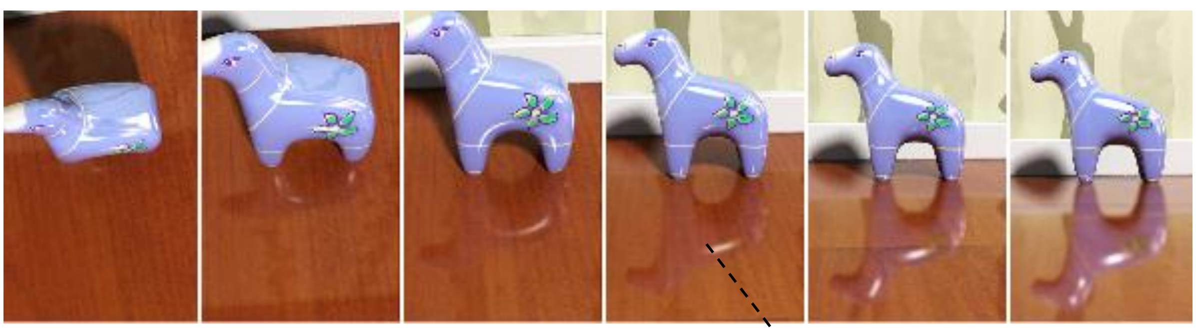
A spectrophotometer, such as the ColorMunki is ideal to capture accurately the color of materials and finishes.

BRDF curve

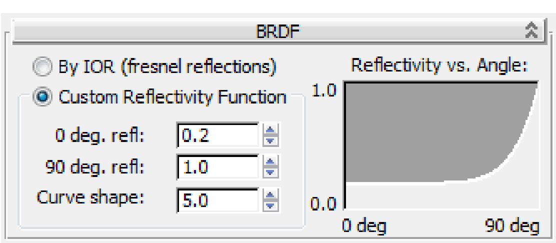
The basics

Besides diffuse color, we need to take into account the “specular” reflectivity amount. This is typically known as the BRDF curve.

If your camera is fixed, the BRDF amount can will give you the illusion that it changes the total amount of reflection, but in fact, that goes hand in hand with the viewing angle.



The amount of reflectivity will vary based on the viewing angle



The BRDF curve will vary the amount of perceived reflection, based on the viewing angle.

The spread of the reflection (its sharpness) will be affected by the glossiness parameter.

This is one of the most important rollout in the 3ds Max A&D material!

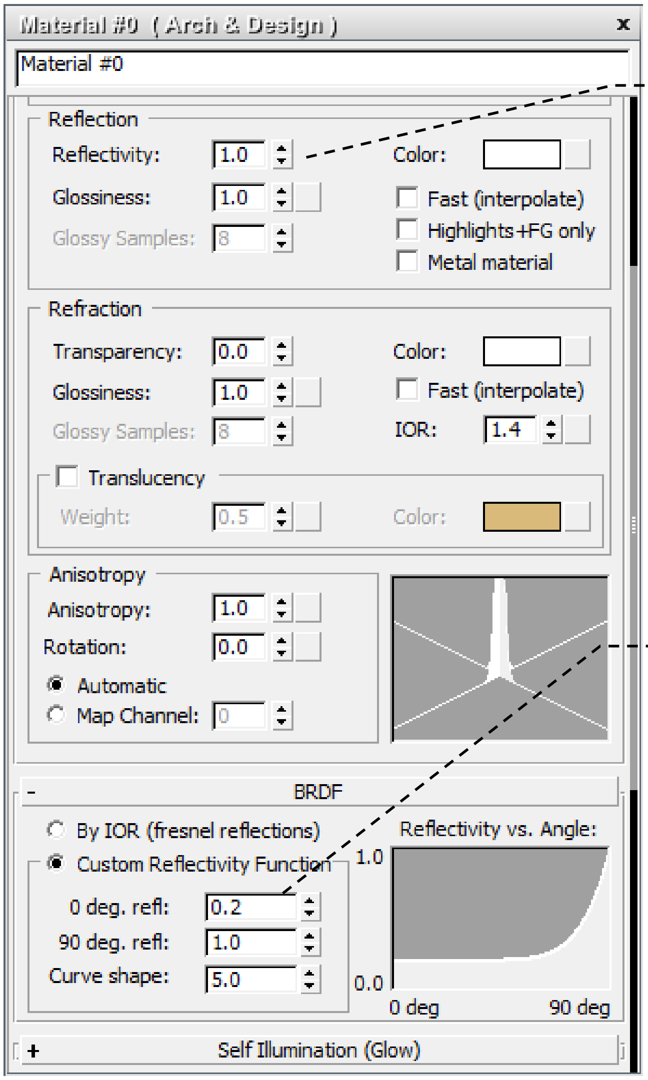
The internal equation of the A&D material

There is an important detail that is often misunderstood about the A&D material. It is the relationship between the Reflection Color, Reflection Amount and BRDF curves. In fact, they are all multiplied together:

Total Reflection Amount (at given angle) = Reflection Color (RGB) * Reflection Amount (float) * BRDF Curve

That means that if you set a reflection color to black, you will get no reflections at all.

My recommendation is to start with the following values:



Leave the reflectivity amount and reflection colors to 100%.

This sounds strange, but this will defer the control to the BRDF parameters….

… then the BRDF values have the total control over the reflectivity @ angle.

In fact, the 3 parameters are mulitplied togheter internally:

Total Reflection Amount (at given angle) =

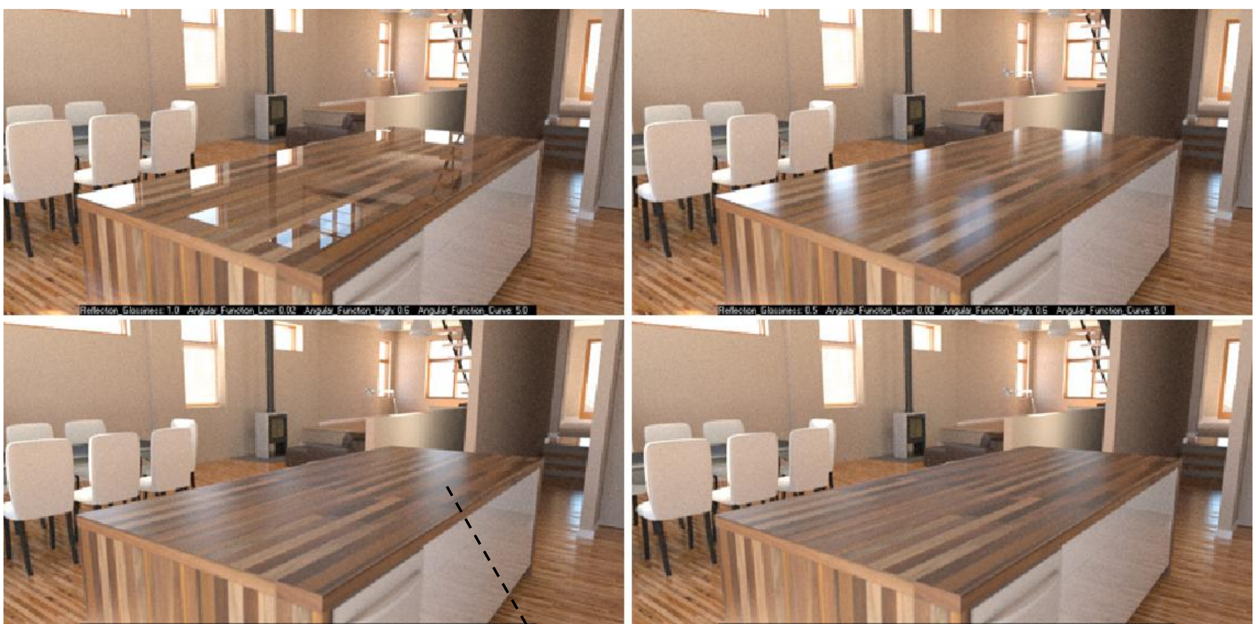
*Reflection Color (RGB) * Reflection Amount (float) * BRDF Curve*

The A&D material

Glossiness

Glossiness represents the spread of reflections. It simply distributes the intensity of the total amount of reflection, as if it was "blurring" it.

Lowering the glossiness will increase the "spread" applied to reflections.



Because this is dependent on the viewing angle, the "spread" is effective only in one direction. It is not like a "blur" which would occur in all directions.

Introducing subtle variations

Bitmap as data

What is wrong here?

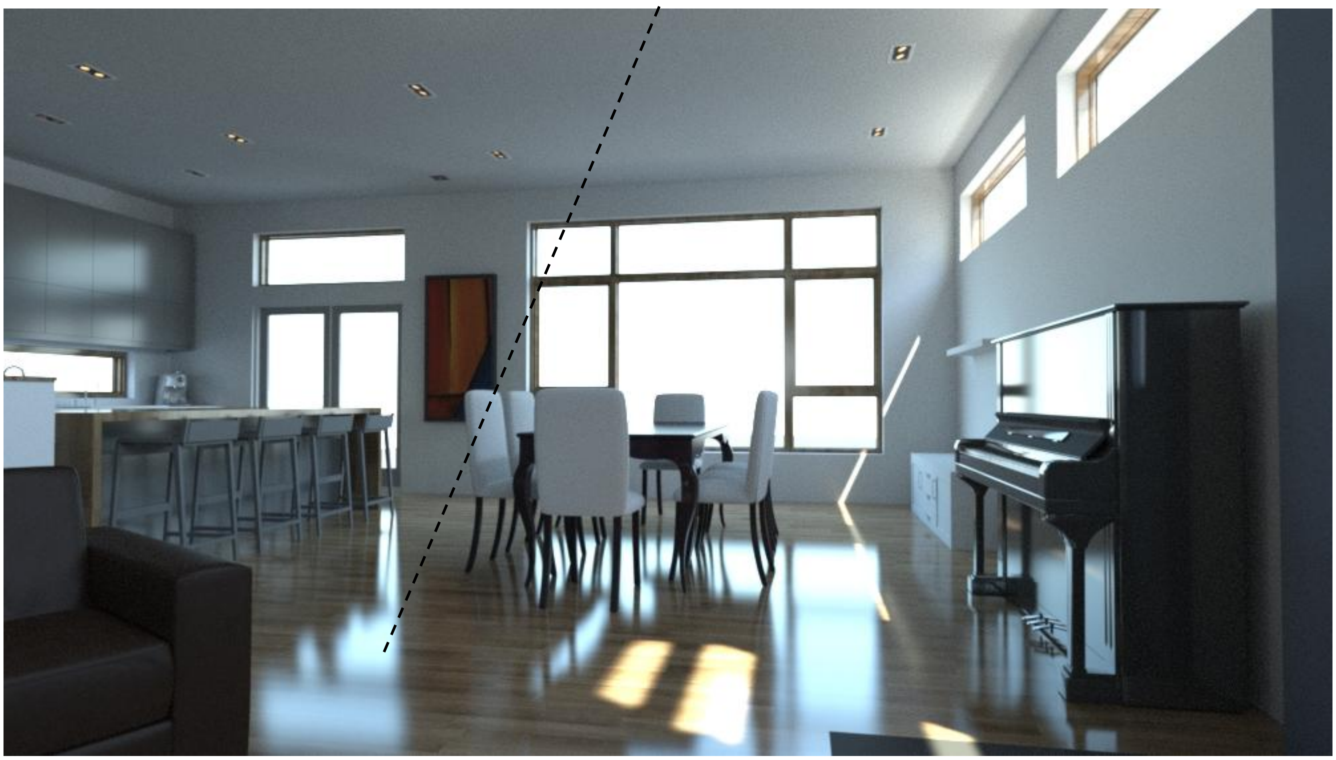
Here is a typical scenario: one puts a nice texture on a floor, set some reflectivity and glossiness and the results are disappointing.

The diffuse map

Meh!

The floor has no defects.

Its perfect and produces a plastic look!



Although the lighting effects are subtle and the image looks relatively good, the floor looks like plastic.

A typical mistake

To solve this, we are often tempted to re-use the bitmaps used in the diffuse channel “as-is”. While this technically works, it is generally not producing great results:

When using the diffuse map in the bump channel, we get perturbations that are based on the color variations of the diffuse map.

In real life, this is rarely the case: deformations are typically independent from the color of materials.

The diffuse map used in the bump channel, with variations on the Bump Amount

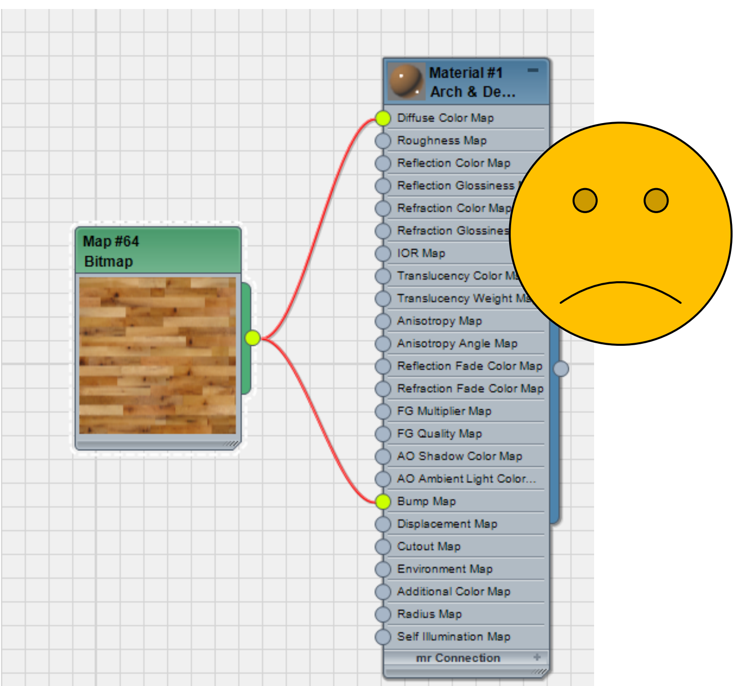
Why is that?

Let’s have a look at a (real) wood floor:

Each plank is angled slightly differently than the next one, producing breakdowns in the reflections.

Those variations are independent from the color.

Therefore, the same bitmap should not be used in the bump channel.



While technically possible, this is generally an artistically wrong practice!

Much better!

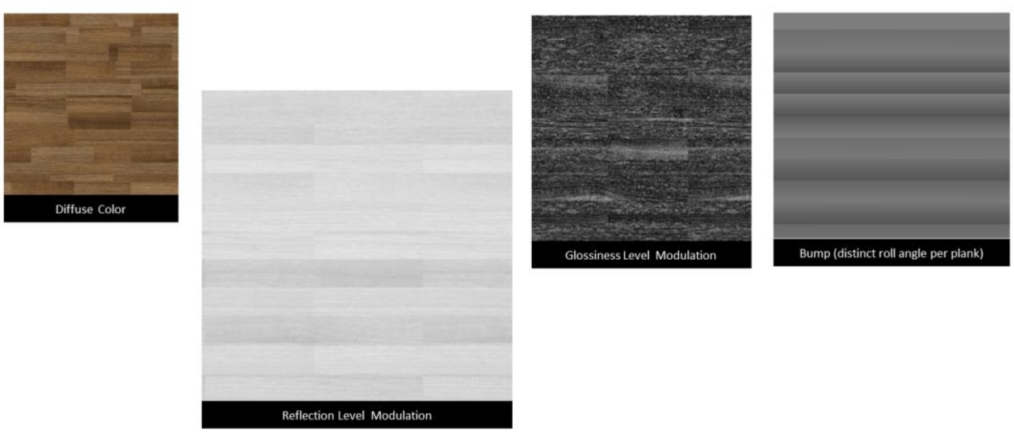
By separating color from perturbations, and using a texture that _actually_ represents real deformations, we get results that are much more realistic. However, this requires to consider bitmap pixels as numeric data as opposed to color information.

For example, this bump map shows gradual ramps representing rotation angle (normal perturbation) for each wood planks:

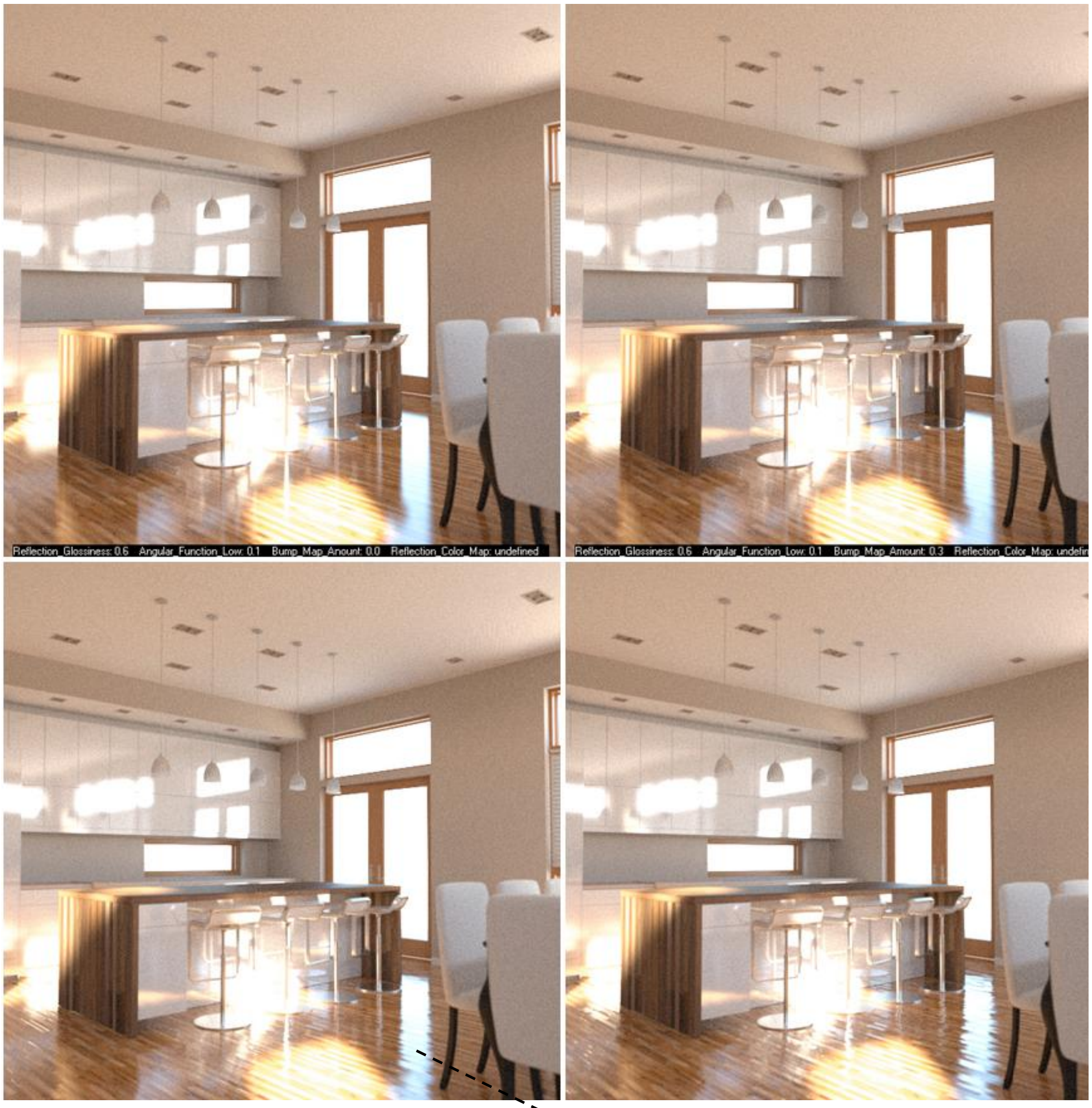
The bump map is representing a slightly different angle for each wood plank.

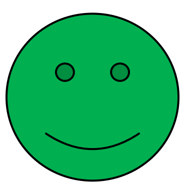
The diffuse map is business as usual, but has the same scale and pattern as the bump map.

You can then consider using the same technique for glossiness, reflection amount, displacement, transparency and so on:



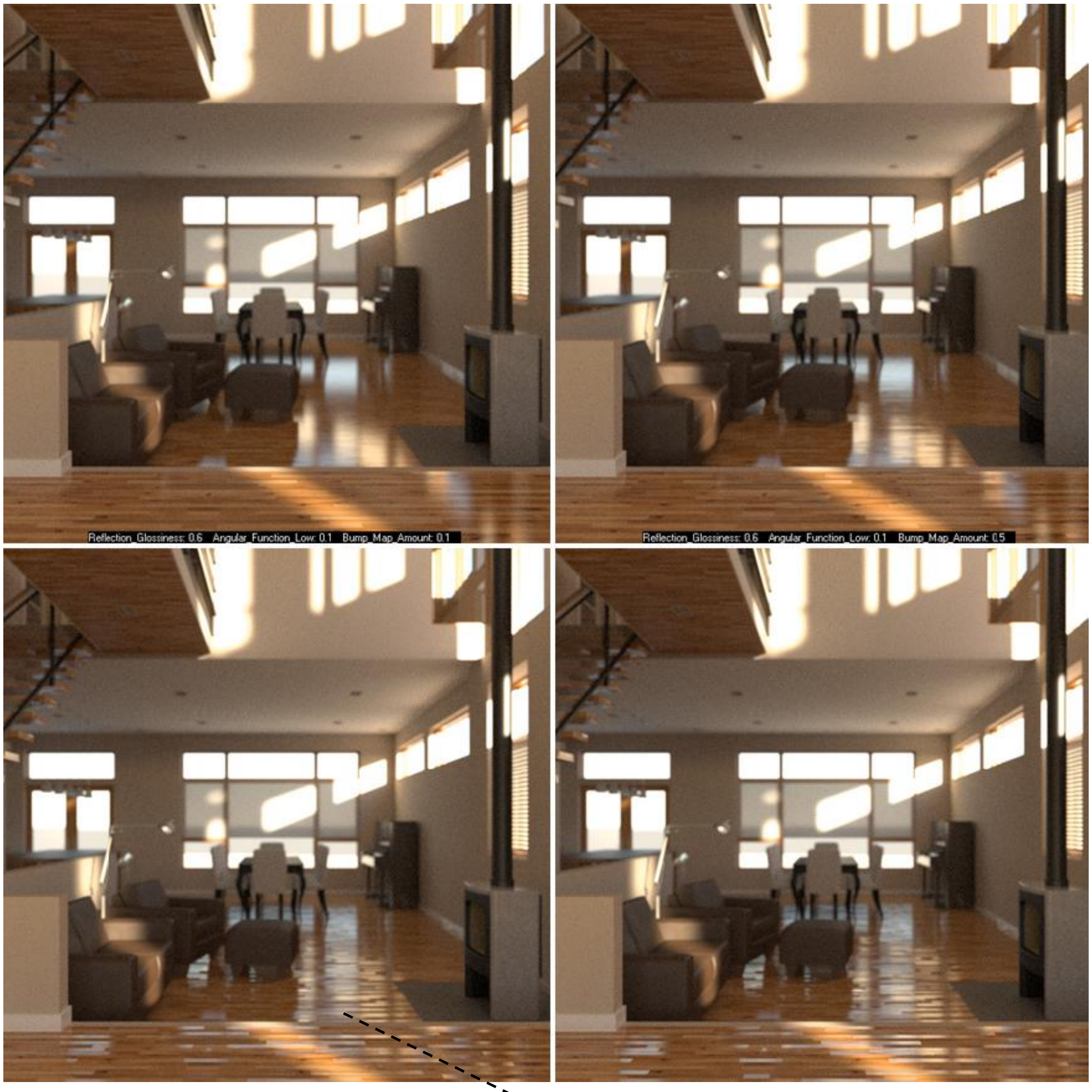
A distinct map for each function, where pixel values represent a number, not a color.





Although the amount of bump can become exagerated (image 4), a separate map representing a distinct orientation for each wood plank produces convincing results!

Two distinct maps, serving distinct purposes feed in the material

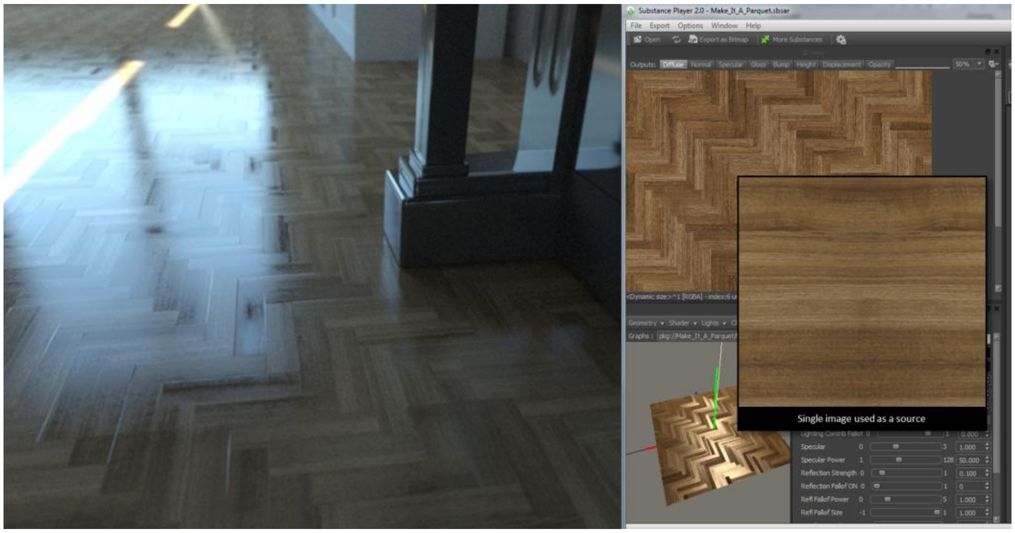


Here, we clearly see that the bump affects the angle of each plank

(exageration done on purpose to visualize the results better)

Automating the process with Allegorithmic Substances

Allegorithmic sells some nifty tools to generate procedurally wood floors and stone walls. Here are some examples:



Make it Parquet substance generates all necessary maps from a single input image

Large scale noise for even more subtle variations

Human made surfaces are rarely perfect, especially on large scale. Only polygons are perfectly flat. As a result, images tend to look a bit too perfect. A good technique consists into using a large scale noise as a bump to breakup reflections a bit better.

Large scale bump, very low amplitude will distort reflections slightly.

A large scale noise is applied as a bump to distort reflections a little bit.

But one problem remains: the “waves” are seamless, even across cabinet doors, which is not possible in reality.

To solve this problem, we can apply a UVW coordinates offset on a couple doors, to break up the pattern:

Each cabinet door and drawer has a slightly offset UVW coordinates to disjoint the large scale noise, resulting in a more natural look