AV10331 - Advanced Material (Appearances) Creation for Rendering

Speaker: Pierre-Felix Breton, Autodesk Design & Visualization

Class Description

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

You will learn tips and techniques to define materials in Autodesk, Inc., software systems 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.

About the Speaker

Pierre-Felix Breton

Pierre-Felix Breton is a user experience designer who specializes in the field of physically-based lighting simulation and rendering.

Employed by Autodesk since 1998, he participates in the creation of core rendering technologies that are embedded into applications such as Autodesk® Revit®, Autodesk® AutoCAD®, Autodesk® Fusion360® and Autodesk® 3ds Max®

His professional background includes electronics, electrical engineering, computer programming, architectural and stage lighting.

Modeled in Revit, rendered in 3ds Max

Contents

Color Management Basics

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

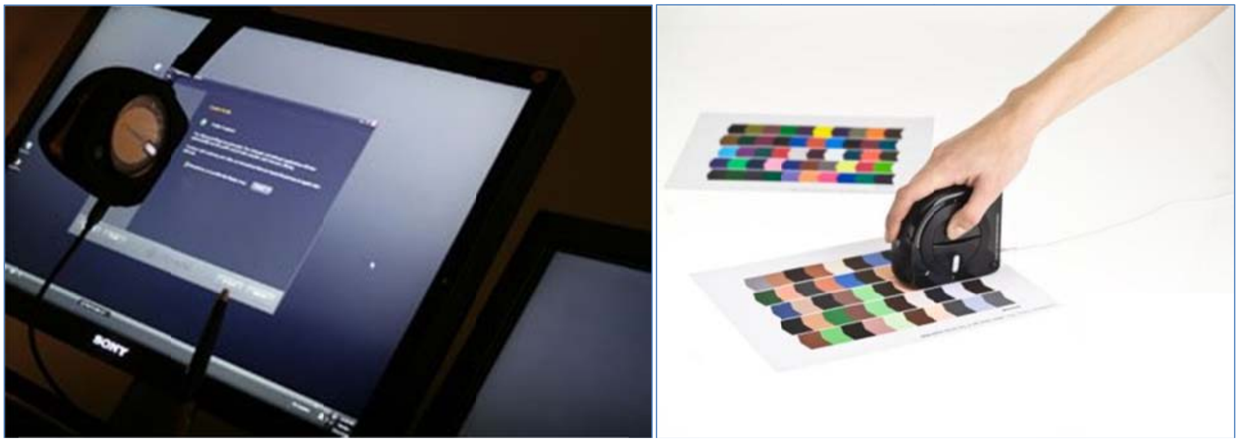
This section provides an overview of the techniques and limitations around color management and calibration. You will learn how to deal with the problem of capturing textures properly, load them in Autodesk® 3ds Max® appropriately and render out images with the correct color settings.

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, this is an excellent starting point!

Step 2: Profile your printer

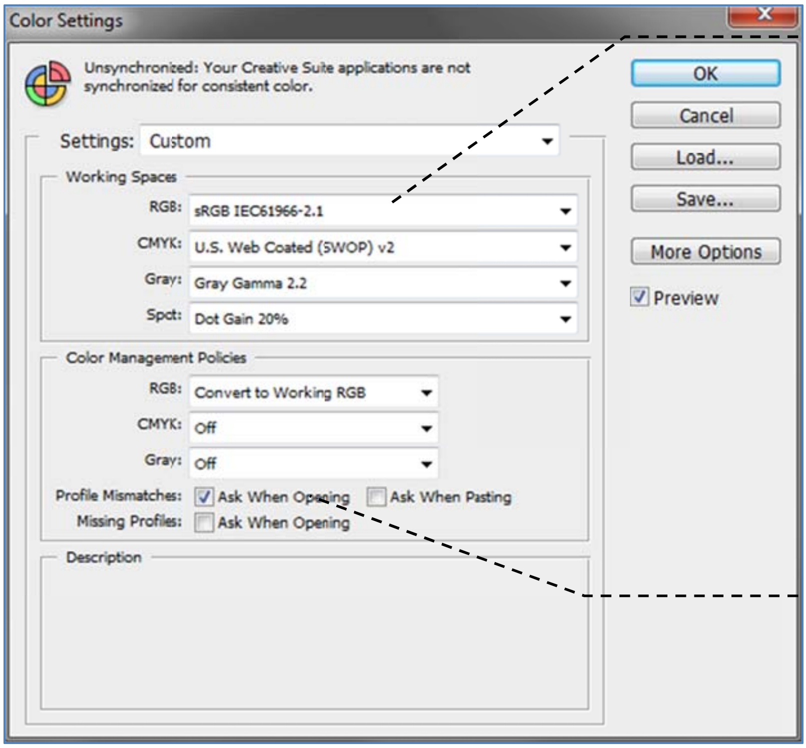
Similarly, the ColorMunki runs you through a wizard that prints a few sample swatches on a color printer. Those swatches are measured back and a color profile is generated. This profile is loaded/installed in Photoshop automatically as well.

Note: Repeat the process for <u>each type of paper</u> you might use: 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.



Color Settings

Unsynchronized: Your Creative Suite applications are not synchronized for consistent color.

Settings: Custom

Working Spaces

RGB: sRGB IEC61966-2.1

CMYK: U.S. Web Coated (SWOP) v2

Gray: Gray Gamma 2.2

Spot: Dot Gain 20%

Color Management Policies

RGB: Convert to Working RGB

CMYK: Off

Gray: Off

Profile Mismatches: ☑ Ask When Opening □ Ask When Pasting

Missing Profiles: □ Ask When Opening

Description

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.

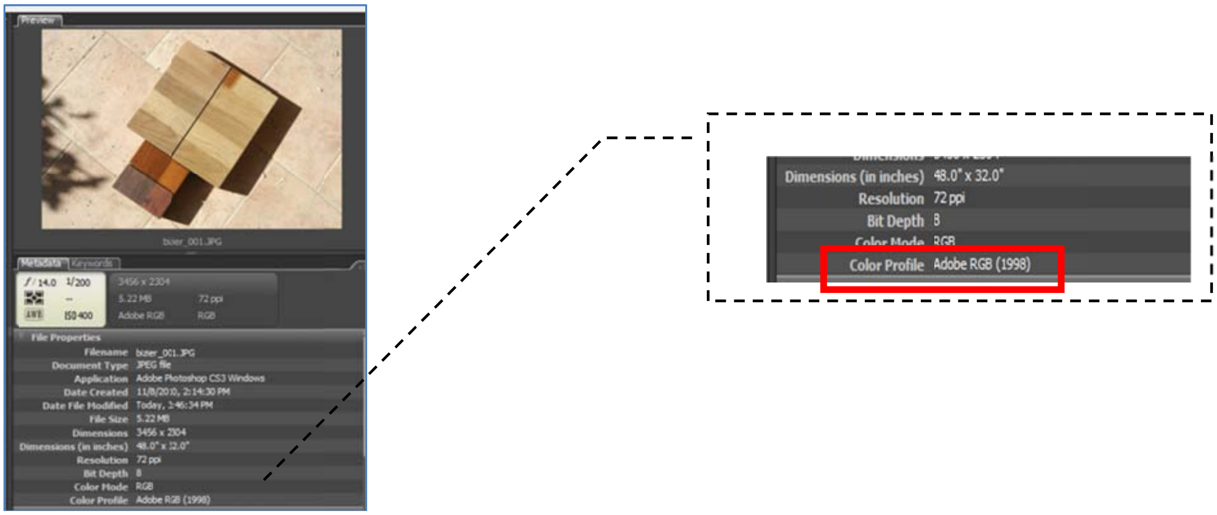
Gotcha!:

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:

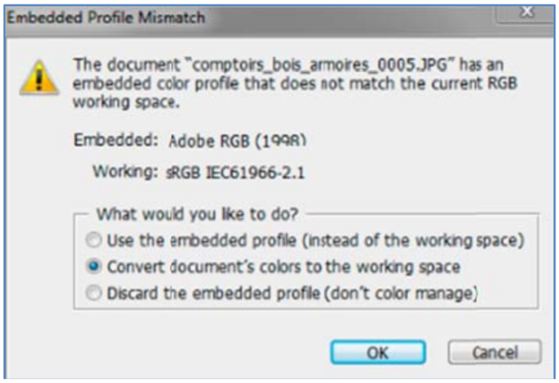


File Properties

Filename	bizer_001.JPG
Document Type	JPEG file
Application	Adobe Photoshop CS3 Windows
Date Created	11/8/2010, 2:14:30 PM
Date File Modified	Today, 3:46:34 PM
File Size	5.22 MB
Dimensions	3456 x 2304
Dimensions (in inches)	48.0" x 32.0"
Resolution	72 ppi
Bit Depth	8
Color Mode	RGB
Color Profile	Adobe RGB (1998)

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:



Embedded Profile Mismatch

The document "comptoirs_bois_armoires_0005.JPG" has an embedded color profile that does not match the current RGB working space.

Embedded: Adobe RGB (1998)

Working: sRGB IEC61966-2.1

What would you like to do?

○ Use the embedded profile (instead of the working space)
● Convert document's colors to the working space
○ Discard the embedded profile (don't color manage)

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

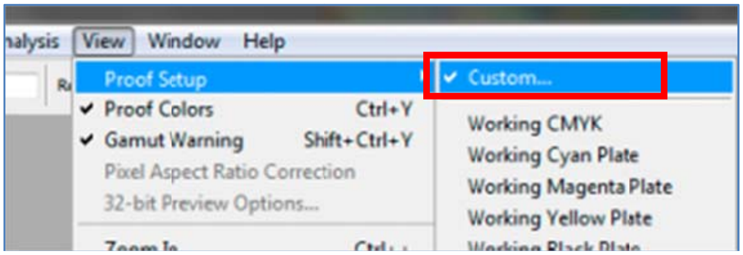
Knowing what each profile does is not really a concern to you in the moment. It is data that is used by Photoshop to convert pixel colors into something that closely matches the color gamut (range) of your display.

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).

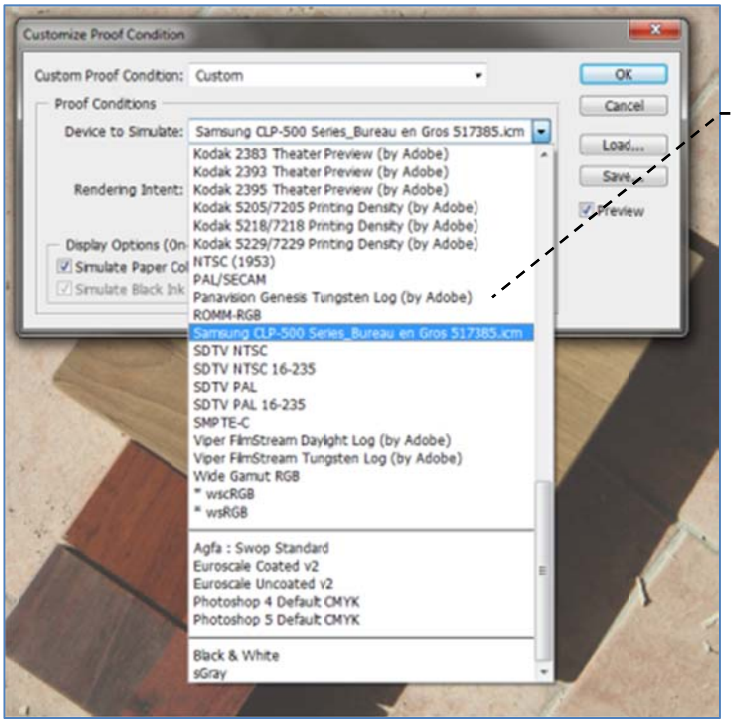
Step 5: Printing from Photoshop

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).


Customize Proof Condition

Custom Proof Condition: Custom

Device to Simulate: Samsung CLP-500 Series_Bureau en Gros 517385.icm

Proof Conditions
Rendering Intent:
Display Options (On-
☑ Simulate Paper Col
☑ Simulate Black Ink
OK
Cancel
Load...
Save...
☑ Preview
Kodak 2383 Theater Preview (by Adobe)
Kodak 2393 Theater Preview (by Adobe)
Kodak 2395 Theater Preview (by Adobe)
Kodak 5205/7205 Printing Density (by Adobe)
Kodak 5218/7218 Printing Density (by Adobe)
Kodak 5229/7229 Printing Density (by Adobe)
NTSC (1953)
PAL/SECAM
Panavision Genesis Tungsten Log (by Adobe)
ROMM-RGB
Samsung CLP-500 Series_Bureau en Gros 517385.icm
SDTV NTSC
SDTV NTSC 16-235
SDTV PAL
SDTV PAL 16-235
SMPTE-C
Viper FilmStream Daylight Log (by Adobe)
Viper FilmStream Tungsten Log (by Adobe)
Wide Gamut RGB
* wscRGB
* wsRGB
Agfa : Swop Standard
Euroscale Coated v2
Euroscale Uncoated v2
Photoshop 4 Default CMYK
Photoshop 5 Default CMYK
Black & White
sGray

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 “Proof 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.

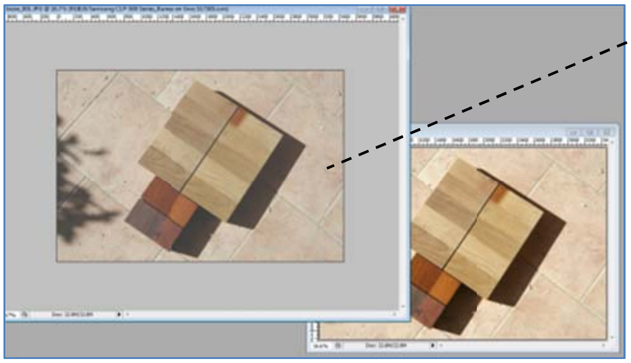
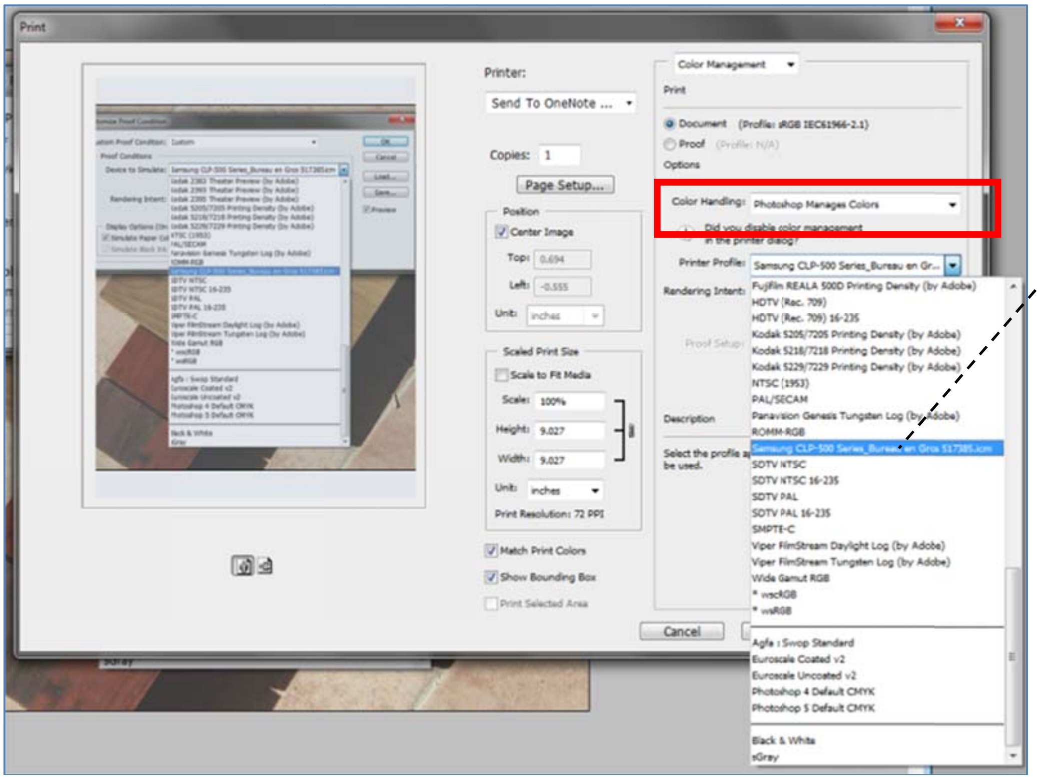


Image seen under "Proof Colors" mode, using a target output matching my printer profile.

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:



Print

Printer: Send To OneNote ...

Copies: 1

Page Setup...

Position

☑ Center Image

Top: 0.694

Left: -0.555

Unit: inches

Scaled Print Size

☐ Scale to Fit Media

Scale: 100%

Height: 9.027

Width: 9.027

Unit: inches

Print Resolution: 72 PPI

☑ Match Print Colors

☑ Show Bounding Box

☐ Print Selected Area

Color Management

Print

● Document (Profile: sRGB IEC61966-2.1)

○ Proof (Profile: N/A)

Options

Color Handling: Photoshop Manages Colors

Did you disable color management in the printer dialog?

Printer Profile: Samsung CLP-500 Series_Bureau en Gr...

Rendering Intent:

Proof Setup:

Description

Select the profile a... be used.

Fujifilm REALA 500D Printing Density (by Adobe)
HDTV (Rec. 709)
HDTV (Rec. 709) 16-235
Kodak 5205/7205 Printing Density (by Adobe)
Kodak 5218/7218 Printing Density (by Adobe)
Kodak 5229/7229 Printing Density (by Adobe)
NTSC (1953)
PAL/SECAM
Panavision Genesis Tungsten Log (by Adobe)
ROMM-RGB
Samsung CLP-500 Series_Bureau en Gros 517385.icm
SDTV NTSC
SDTV NTSC 16-235
SDTV PAL
SDTV PAL 16-235
SMPTE-C
Viper FilmStream Daylight Log (by Adobe)
Viper FilmStream Tungsten Log (by Adobe)
Wide Gamut RGB
* wscRGB
* wsRGB
Agfa : Swop Standard
Euroscale Coated v2
Euroscale Uncoated v2
Photoshop 4 Default CMYK
Photoshop 5 Default CMYK
Black & White
sGray
Cancel

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

Photoshop Print Configuration Dialog with Color Management enabled.

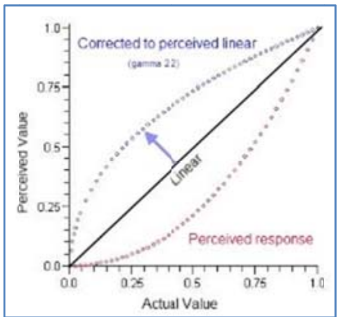
The resulting print output will be as close as possible to the image you saw on screen. Trying it is adopting it ☺

The importance of Gamma correction in rendering

Display encoding vs. Data encoding

Introduction to Gamma correction

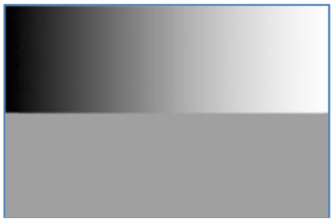
Display systems have their own inherent characteristics that are not perceptually uniform. Typically, images displayed "as-is" appear incorrect to the human eye. To counteract this, an adjustment is introduced. This adjustment is known as "*Gamma Correction*".

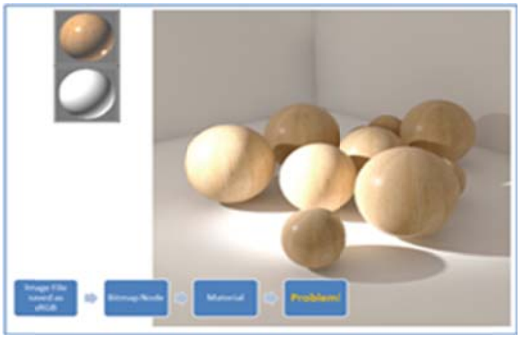


Gamma correction graph

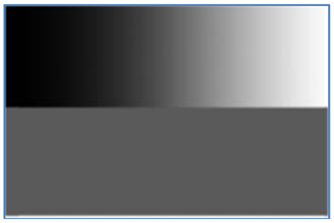
But here is the dilemma: images must be gamma corrected for display purposes (so they *look good*), but this creates a situation where physically incorrect data can be feed to 3D rendering engines. Typically, this problem occurs with diffuse colors for materials.

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

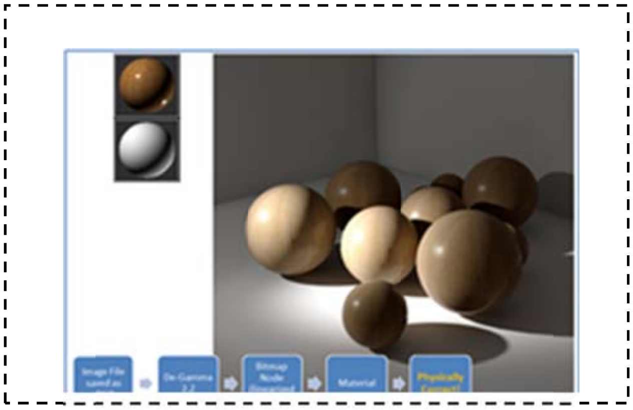




Gamma Corrected 50% grey:
Appear correct on display,
feed ***incorrect*** *(too bright) reflectance* ***data*** *to 3D renderers*



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"**.

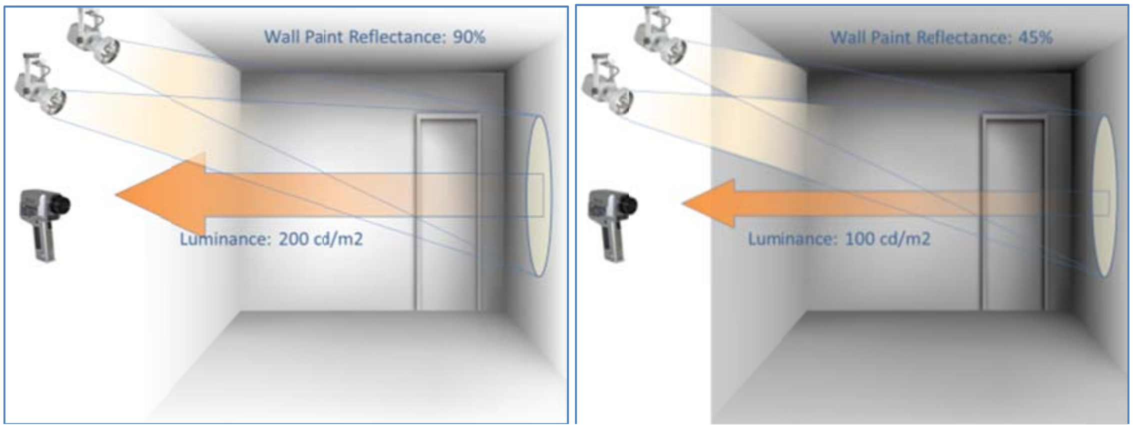
In summary, images used as <u>data to perform calculation</u> – or used to feed a numerical value to the renderer (diffuse textures, bump maps, normal maps) must be in *linear space* from the point of view of the 3D renderer while images used for display must be gamma corrected to look good (i.e. *non-linear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

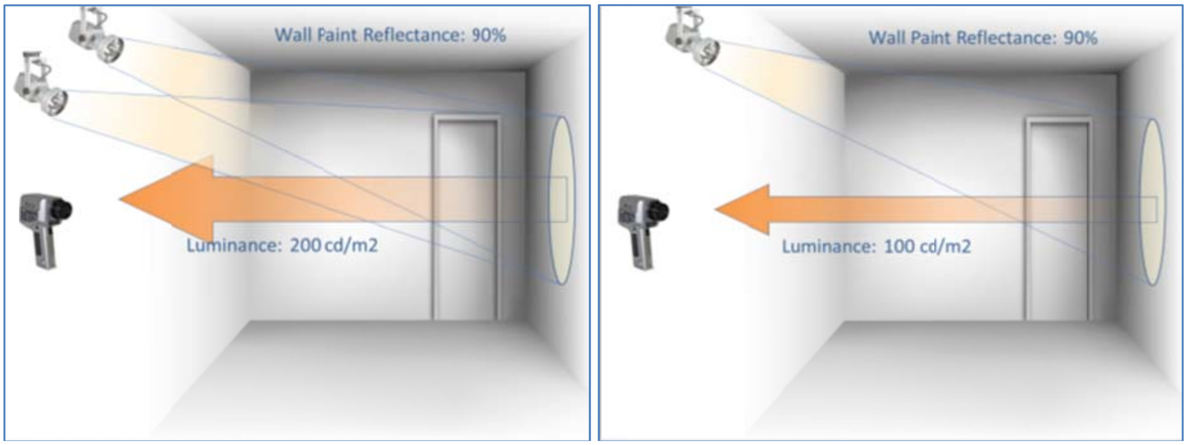
Light is **MULTIPLIED** with the reflectance (color) of the wall:



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*.

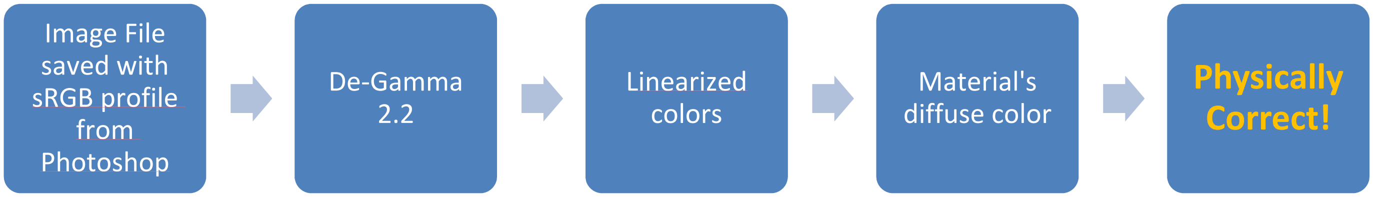
3D Renderers calculate in **LINEAR SPACE**.

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.

Gamma settings for Revit, AutoCAD, Fusion 360:

Gamma handling is done automatically. In other words, you don't have to make special steps in order to have renderings as physically correct as possible.

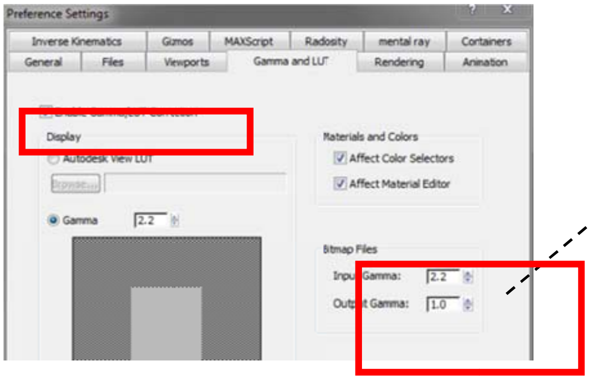
Gamma settings for 3ds Max (version 2014 and above):

This workflow is now handled automatically. Generally, you don't have to bother about making anything special in 3ds max. 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".

Gamma settings for 3ds Max (version 2013 and lower):

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



Display

○ Autodesk View LUT

● Gamma 2.2

Materials and Colors

☑ Affect Color Selectors

☑ Affect Material Editor

Bitmap Files

Input Gamma: 2.2

Output Gamma: 1.0

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

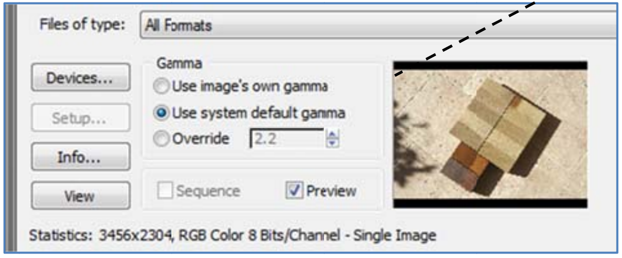
Leave at 2.2 for diffuse textures.

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

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).



Files of type: All Formats

Gamma

○ Use image's own gamma

● Use system default gamma

○ Override 2.2

☐ Sequence ☑ Preview

Statistics: 3456x2304, RGB Color 8 Bits/Channel - Single Image

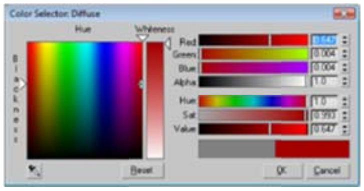
3ds Max: Image File Browse Dialog: this is where you can define a” per bitmap” Gamma correction value.

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.



Color Selector: Diffuse

Red 0.647
Green 0.004
Blue 0.004
Alpha 1.0
Hue 1.0
Sat 0.993
Value 0.647

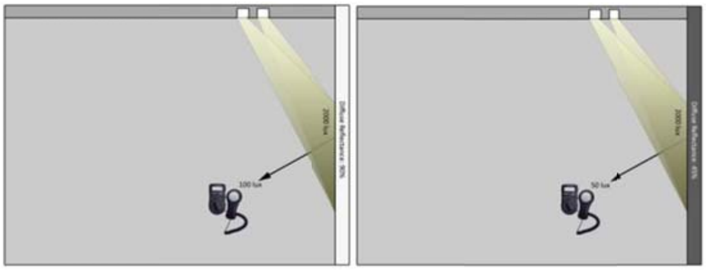
What is the right red to pick for this apple? It can be difficult to tell and trust what you see on screen.

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).

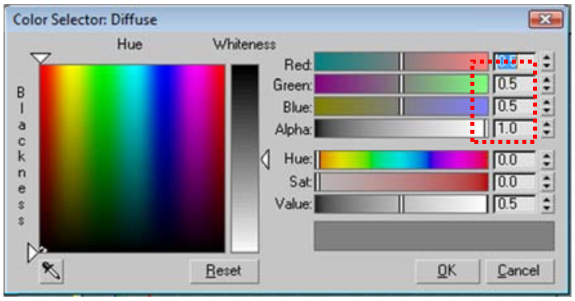
All indirect light bounces off surfaces in a diffuse manner in all directions (also known as “Lambertian”). In contrast, directed bouncing light, such as the pattern of a mirror in sunlight, is called a *caustic or specular reflection.*

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’.

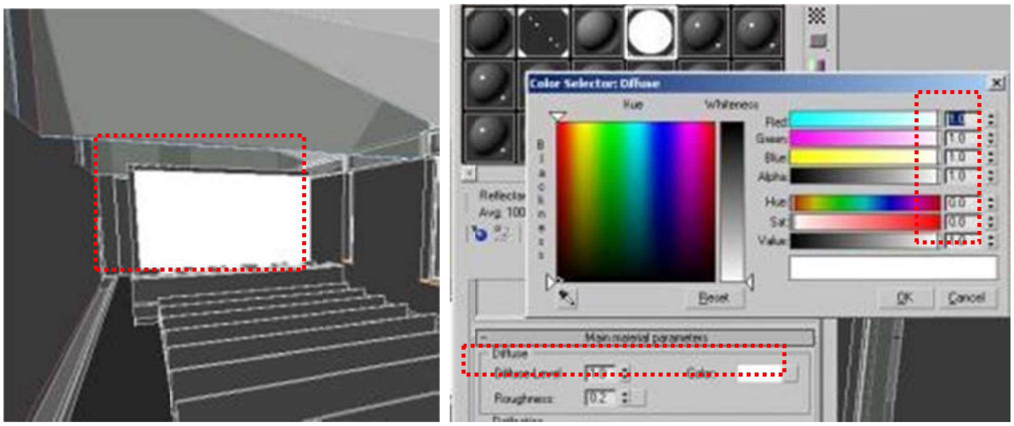
For example, a white wall will not reflect more than about 90% of the light it receives. Therefore, the RGB color should not be more than 0.90, 0.90, 0.90 (floating point colors) or 230, 230, 230 (integer colors).

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.


Color Selector: Diffuse

Red: 0.5
Green: 0.5
Blue: 0.5
Alpha: 1.0
Hue: 0.0
Sat: 0.0
Value: 0.5

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.


Color Selector: Diffuse

Red: 1.0
Green: 1.0
Blue: 1.0
Alpha: 1.0
Hue: 0.0
Sat: 0.0

Main material parameters
Diffuse
Roughness: 0.2

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

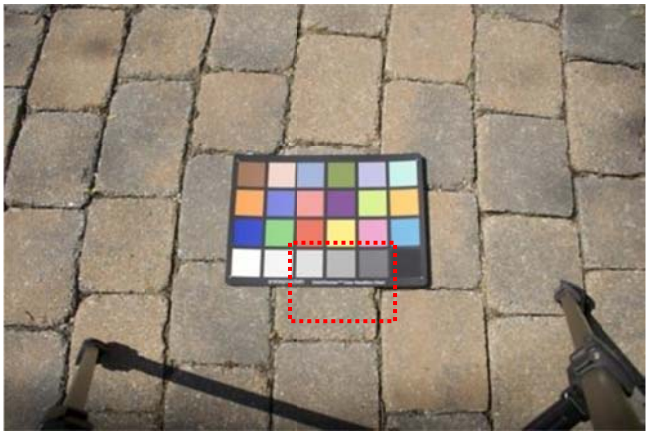
Using a reference chart

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^{TM} 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^{TM} 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).

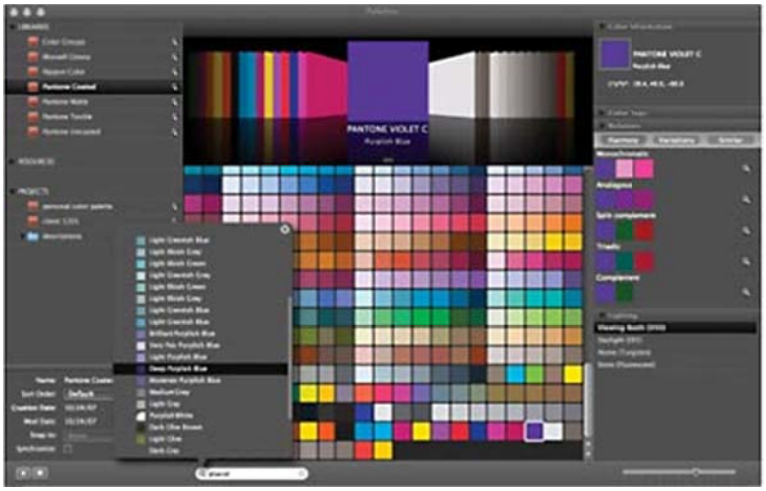


By eye balling, we deduct that the average reflectance of the pavement is roughly 20%: its brightness is close to the swatch #22 on the chart which has a known reflectance of 20%.

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.

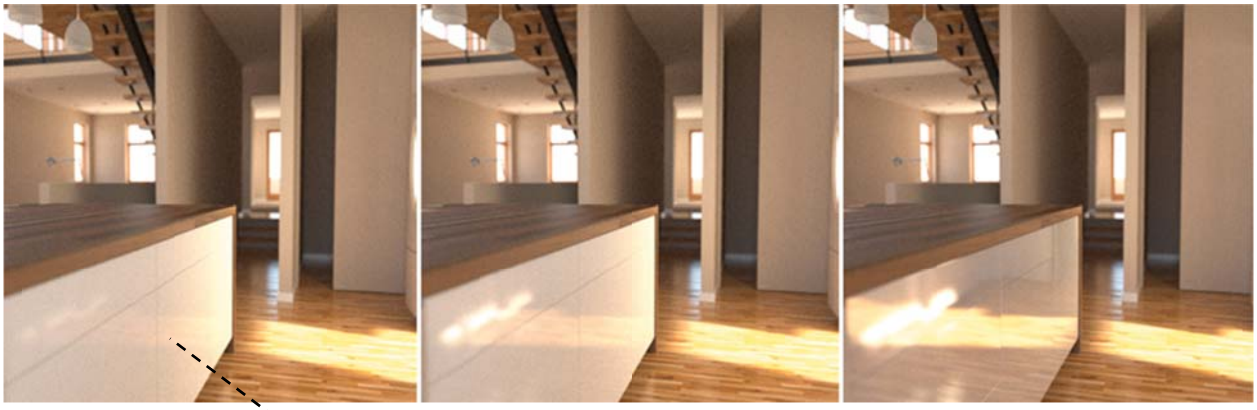
A spectrophotometer, such as the ColorMunki is ideal to capture accurately the color of materials and finishes.



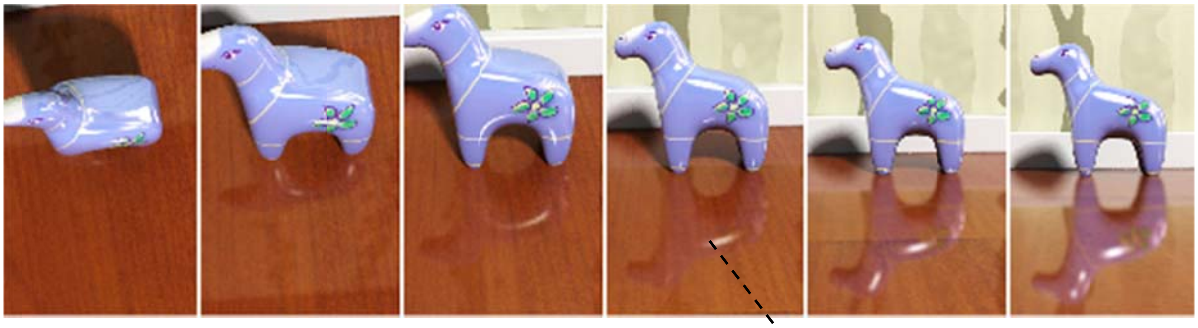
BRDF curve

The basics

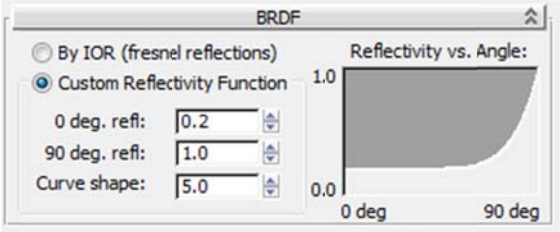
Besides diffuse color, we need to take into account the “specular” reflectivity amount. This is typically known as the BRDF curve.



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



BRDF

○ By IOR (fresnel reflections)

● Custom Reflectivity Function

0 deg. refl: 0.2

90 deg. refl: 1.0

Curve shape: 5.0

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!

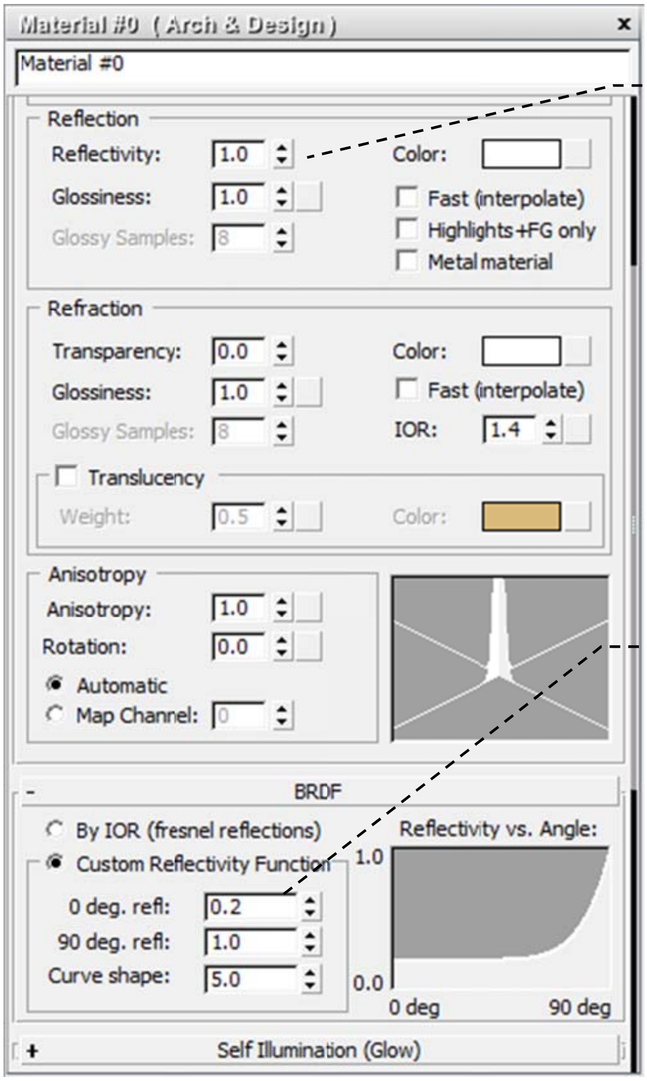
The internal equation of the 3ds Max A&D material

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:



Material #0 (Arch & Design)

Material #0

Reflection

Reflectivity: 1.0 Color:

Glossiness: 1.0 □ Fast (interpolate)

Glossy Samples: 8 □ Highlights+FG only

□ Metal material

Refraction

Transparency: 0.0 Color:

Glossiness: 1.0 □ Fast (interpolate)

Glossy Samples: 8 IOR: 1.4

□ Translucency

Weight: 0.5 Color:

Anisotropy

Anisotropy: 1.0

Rotation: 0.0

● Automatic

○ Map Channel: 0

- BRDF

○ By IOR (fresnel reflections)

● Custom Reflectivity Function

0 deg. refl: 0.2

90 deg. refl: 1.0

Curve shape: 5.0

The A&D material

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.

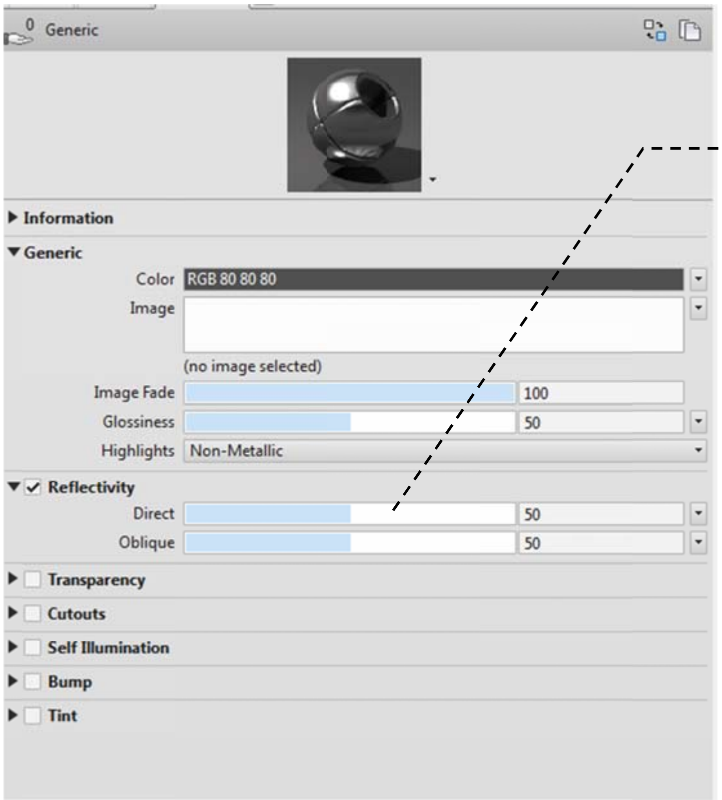
In fact, the 3 parameters are mulitplied togheter internally:

Total Reflection Amount (at given angle) =

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

The internal equation of the Autodesk Generic material (Revit, AutoCAD, Inventor)

The Autodesk Generic material is derived from the 3ds Max A&D material. The interface is simpler, but the principles are the same:



Generic

▶ Information

▼ Generic

Color: RGB 80 80 80

Image: (no image selected)

Image Fade: 100

Glossiness: 50

Highlights: Non-Metallic

▼ ☑ Reflectivity

Direct: 50

Oblique: 50

▶ □ Transparency

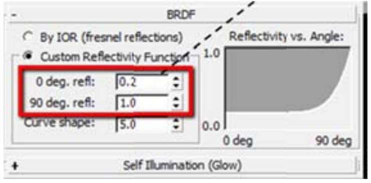
▶ □ Cutouts

▶ □ Self Illumination

▶ □ Bump

▶ □ Tint

This is identical to the A&D material controls. It actually drives the same parameters under the hood:



BRDF

○ By IOR (fresnel reflections)

● Custom Reflectivity Function

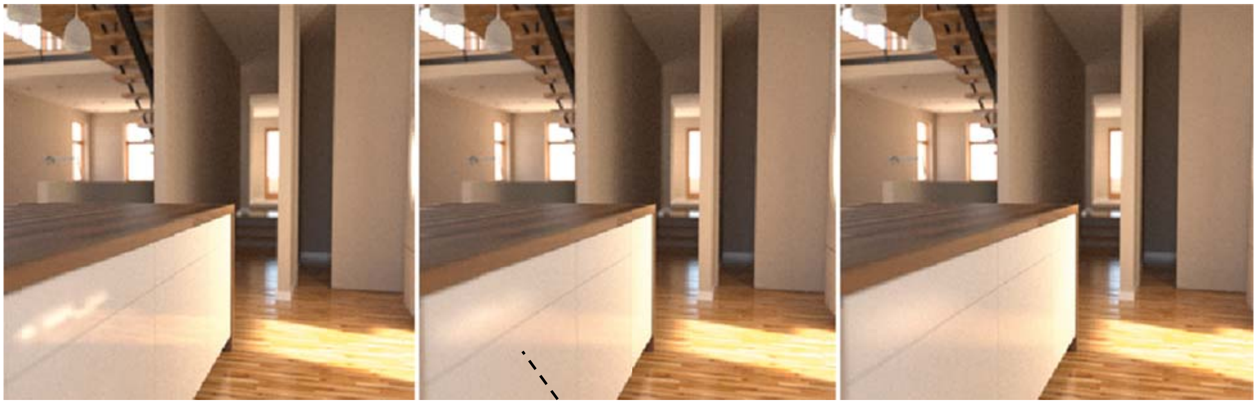
0 deg. refl: 0.2

90 deg. refl: 1.0

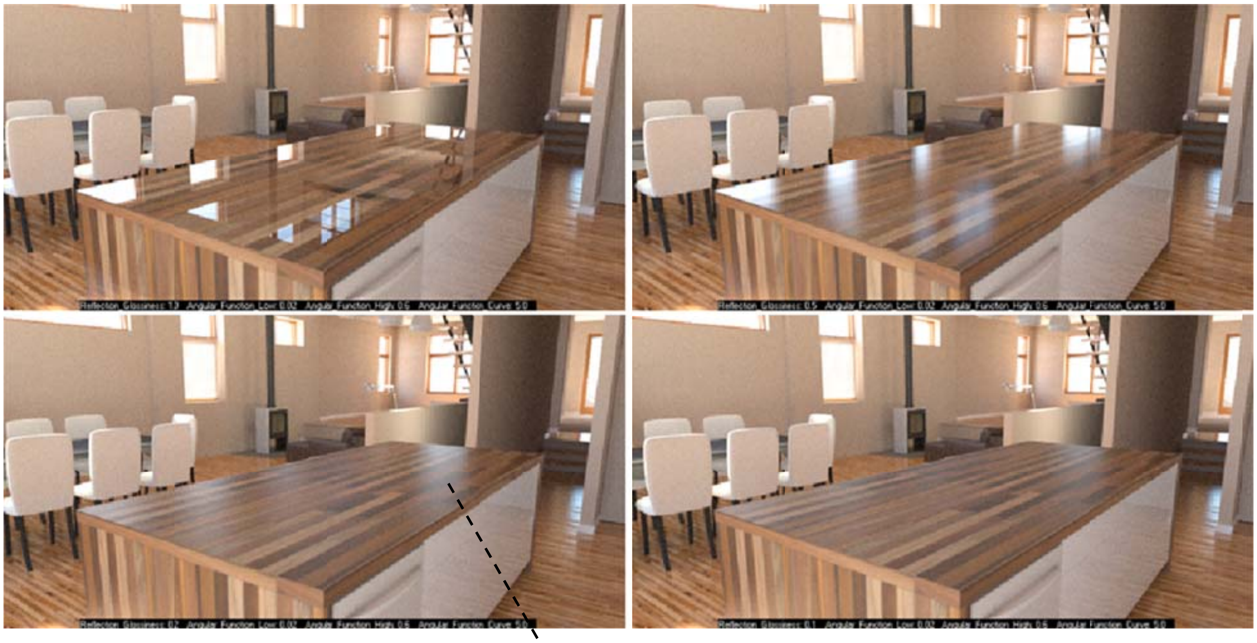
Curve shape: 5.0

Glossiness/Roughness

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.


Reflection_Glossiness: 1.0 Angular_Function_Low: 0.02 Angular_Function_High: 0.6 Angular_Function_Curve: 5.0
Reflection_Glossiness: 0.5 Angular_Function_Low: 0.02 Angular_Function_High: 0.6 Angular_Function_Curve: 5.0
Reflection_Glossiness: 0.2 Angular_Function_Low: 0.02 Angular_Function_High: 0.6 Angular_Function_Curve: 5.0
Reflection_Glossiness: 0.1 Angular_Function_Low: 0.02 Angular_Function_High: 0.6 Angular_Function_Curve: 5.0

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?

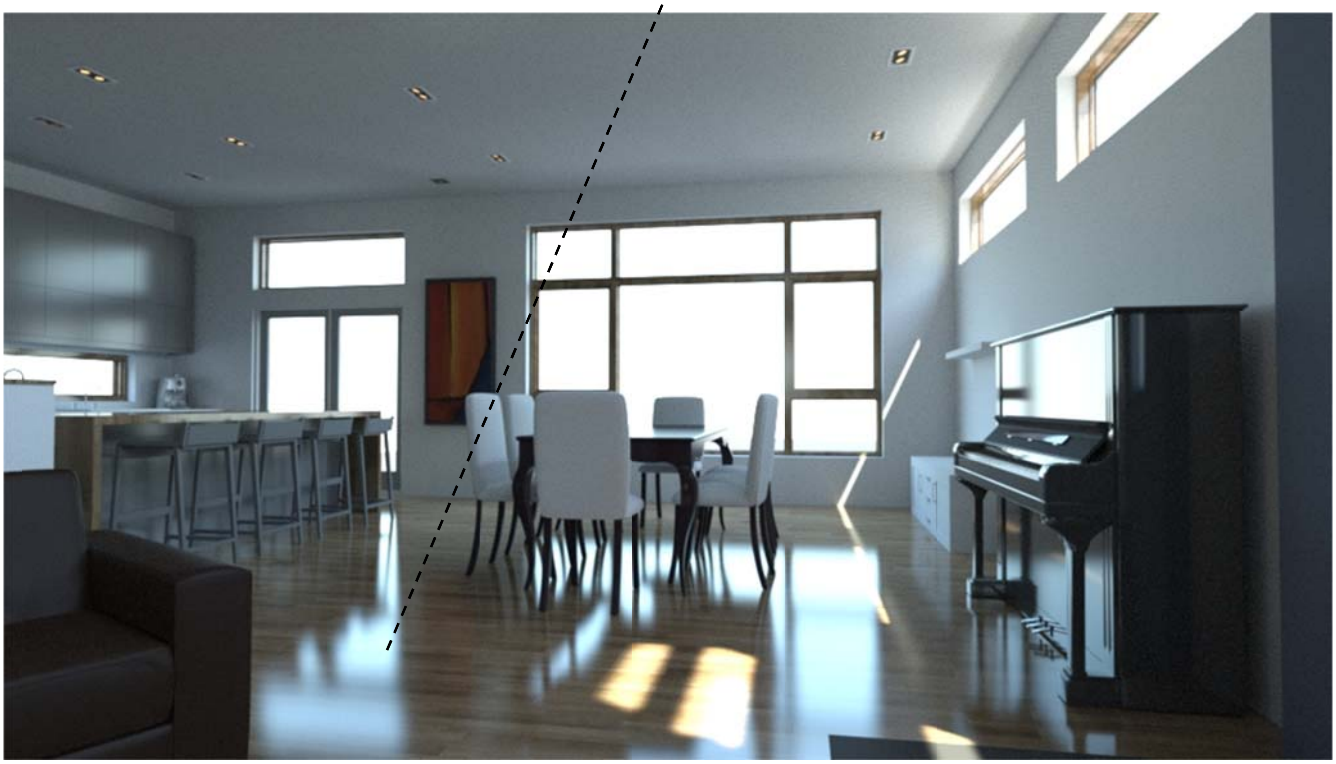
Here is a typical scenario: one puts a nice texture on a floor, set some reflectivity and glossiness and the results are disappointing.

The diffuse map

Meh!

The floor has no defects.

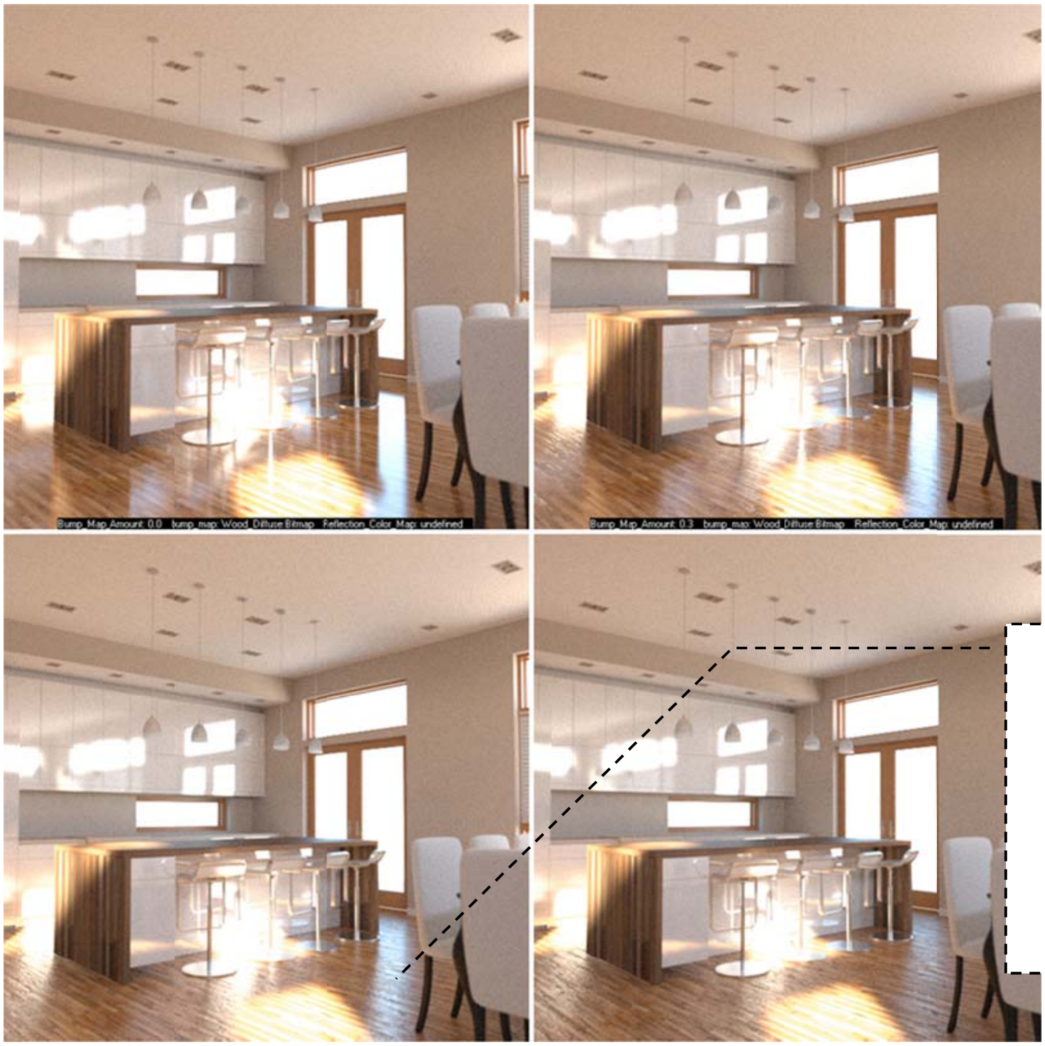
Its perfect and produces a plastic look!



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:


Bump_Map_Amount: 0.0   bump_map: Wood_Diffuse:Bitmap   Reflection_Color_Map: undefined
Bump_Map_Amount: 0.3   bump_map: Wood_Diffuse:Bitmap   Reflection_Color_Map: undefined

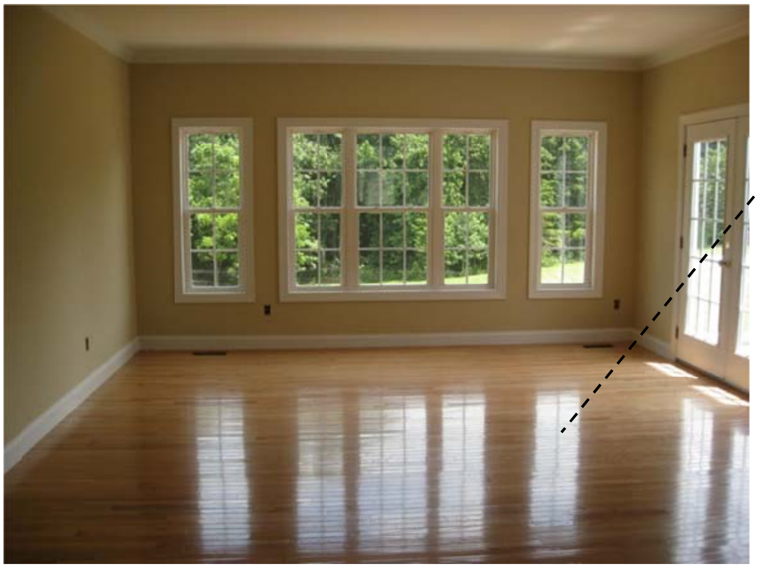
When using the diffuse map in the bump channel, we get perturbations that are based on the color variations of the diffuse map.

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?

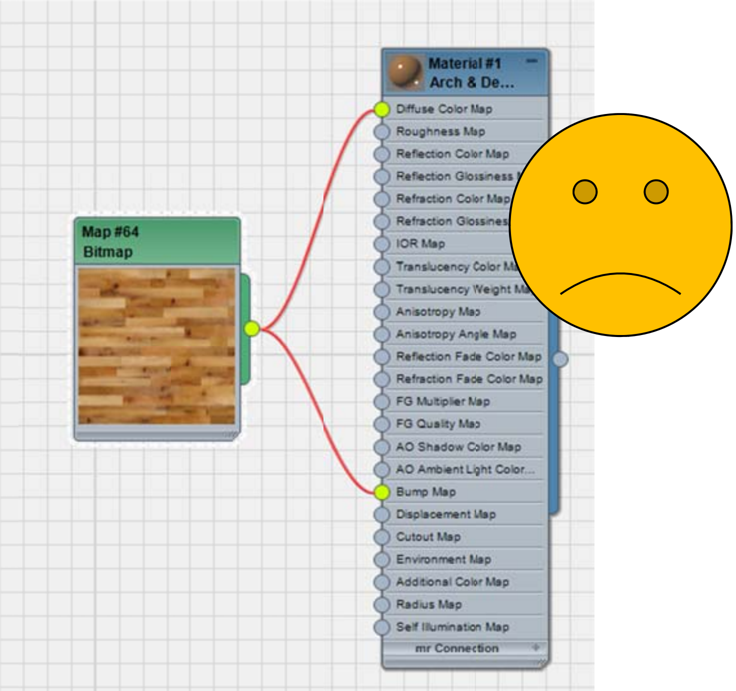
Let’s have a look at a (real) wood floor:



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.

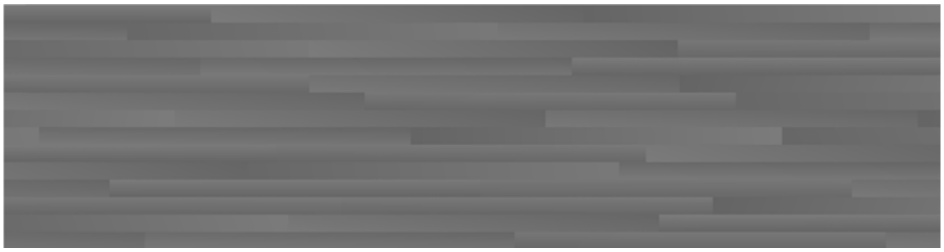


While technically possible, this is generally an artistically wrong practice!

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 considering bitmap pixels as numeric data as opposed to color information.

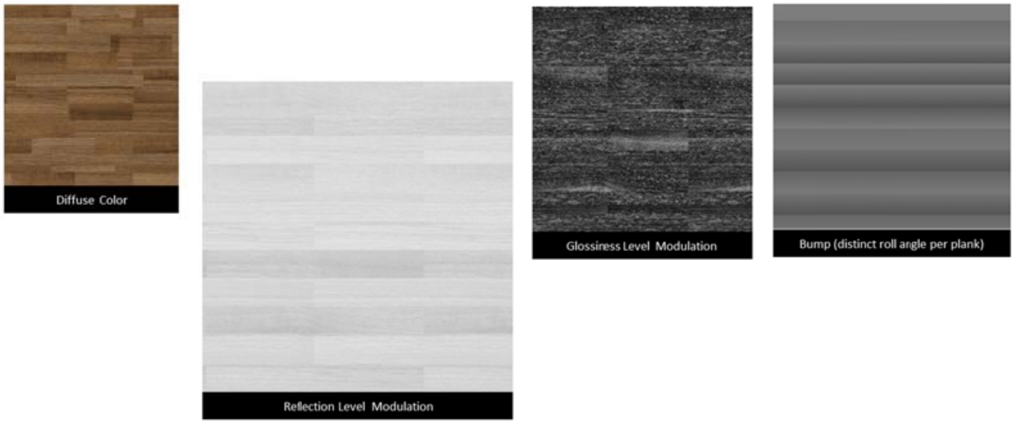
For example, this bump map shows gradual ramps representing rotation angle (normal perturbation) for each wood planks:



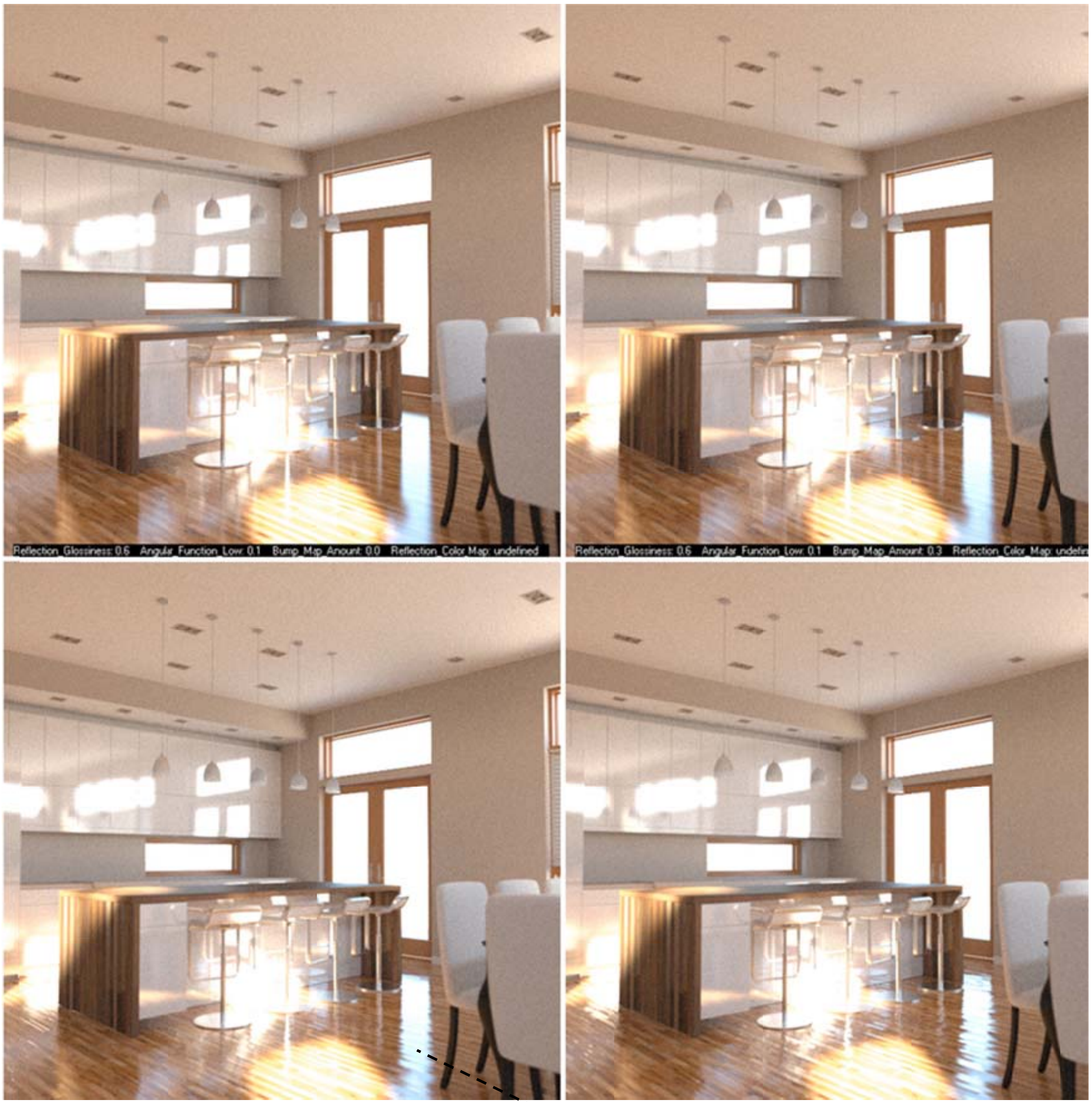
The bump map is representing a slightly different angle for each wood plank.

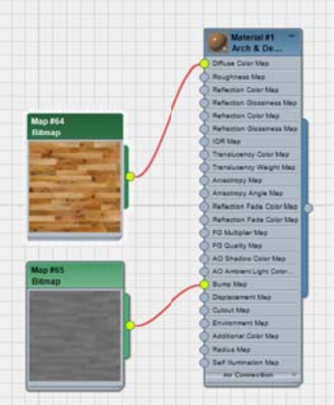
The diffuse map is business as usual, but has the same scale and pattern as the bump map.

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


Diffuse Color
Reflection Level Modulation
Glossiness Level Modulation
Bump (distinct roll angle per plank)

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

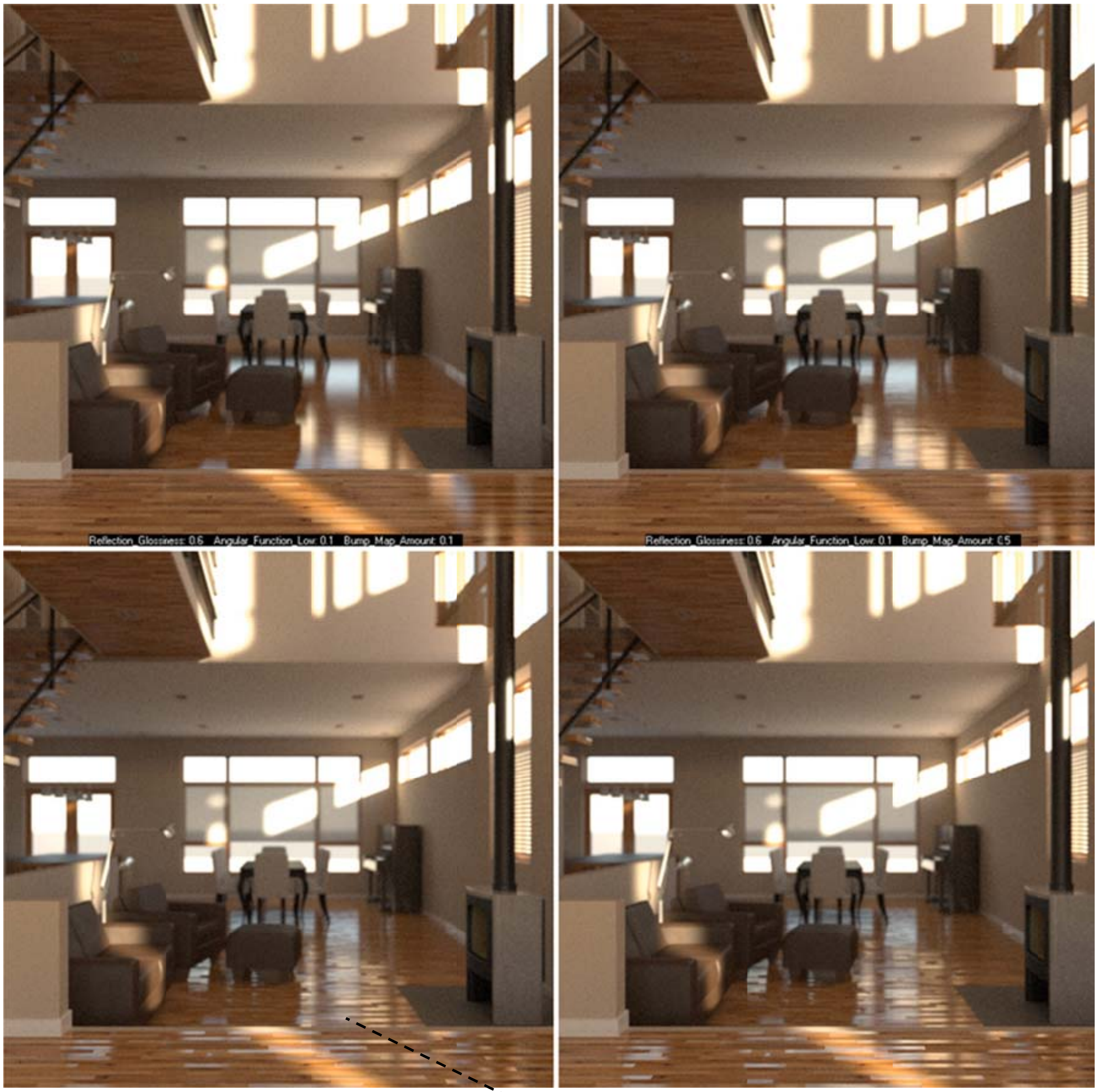

Reflection_Glossiness: 0.6   Angular_Function_Low: 0.1   Bump_Map_Amount: 0.0   Reflection_Color_Map: undefined
Reflection_Glossiness: 0.6   Angular_Function_Low: 0.1   Bump_Map_Amount: 0.3   Reflection_Color_Map: undefin



☺

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



Reflection_Glossiness: 0.6   Angular_Function_Low: 0.1   Bump_Map_Amount: 0.1

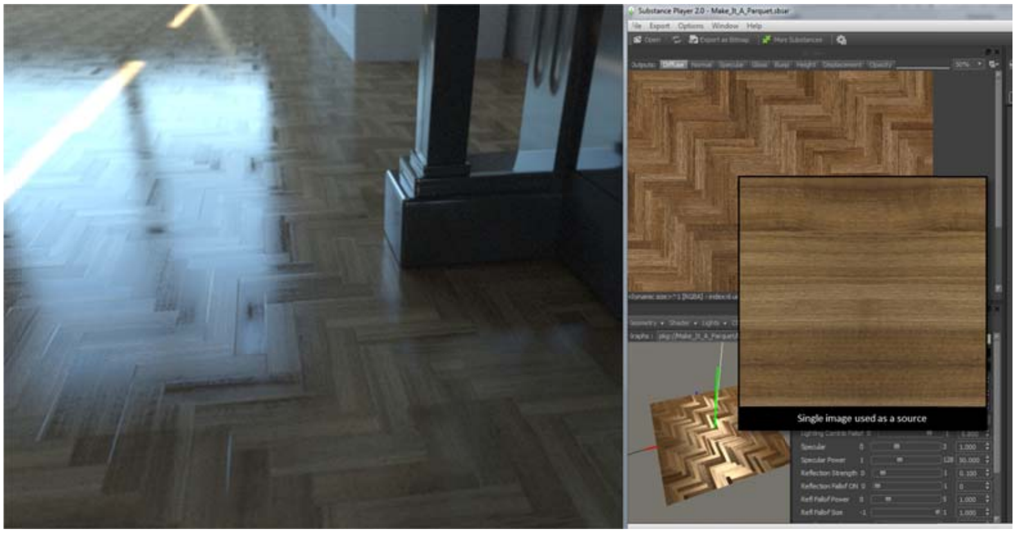
Reflection_Glossiness: 0.6   Angular_Function_Low: 0.1   Bump_Map_Amount: 0.5

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

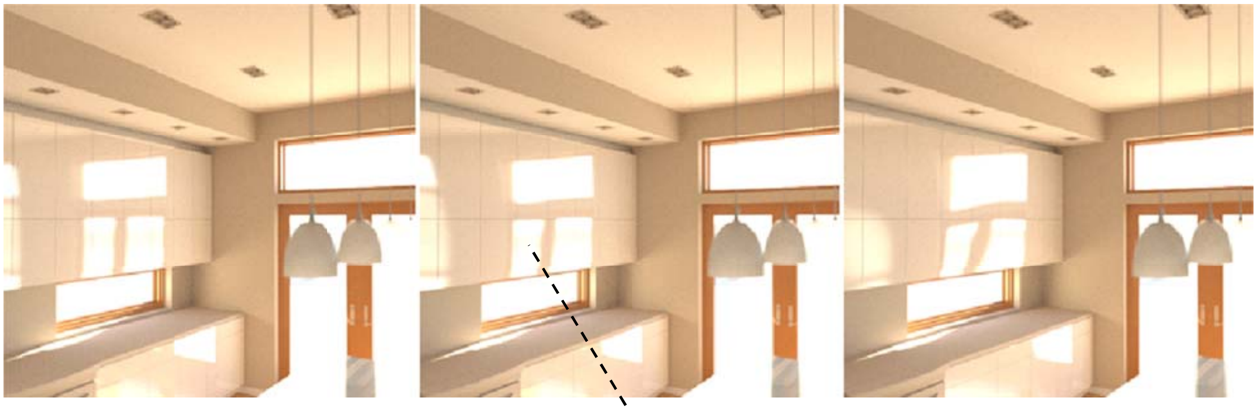
Allegorithmic sells some nifty tools to generate procedurally wood floors and stone walls. Here are some examples:


Substance Player 2.0 - Make_It_A_Parquet.sbsar
File Export Options Window Help
Open Export as Bitmap More Substances
Outputs: Diffuse Normal Specular Gloss Bump Height Displacement Opacity
Single image used as a source
Specular
Specular Power
Reflection Strength
Reflection Falloff ON
Refl Falloff Power
Refl Falloff Size

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.

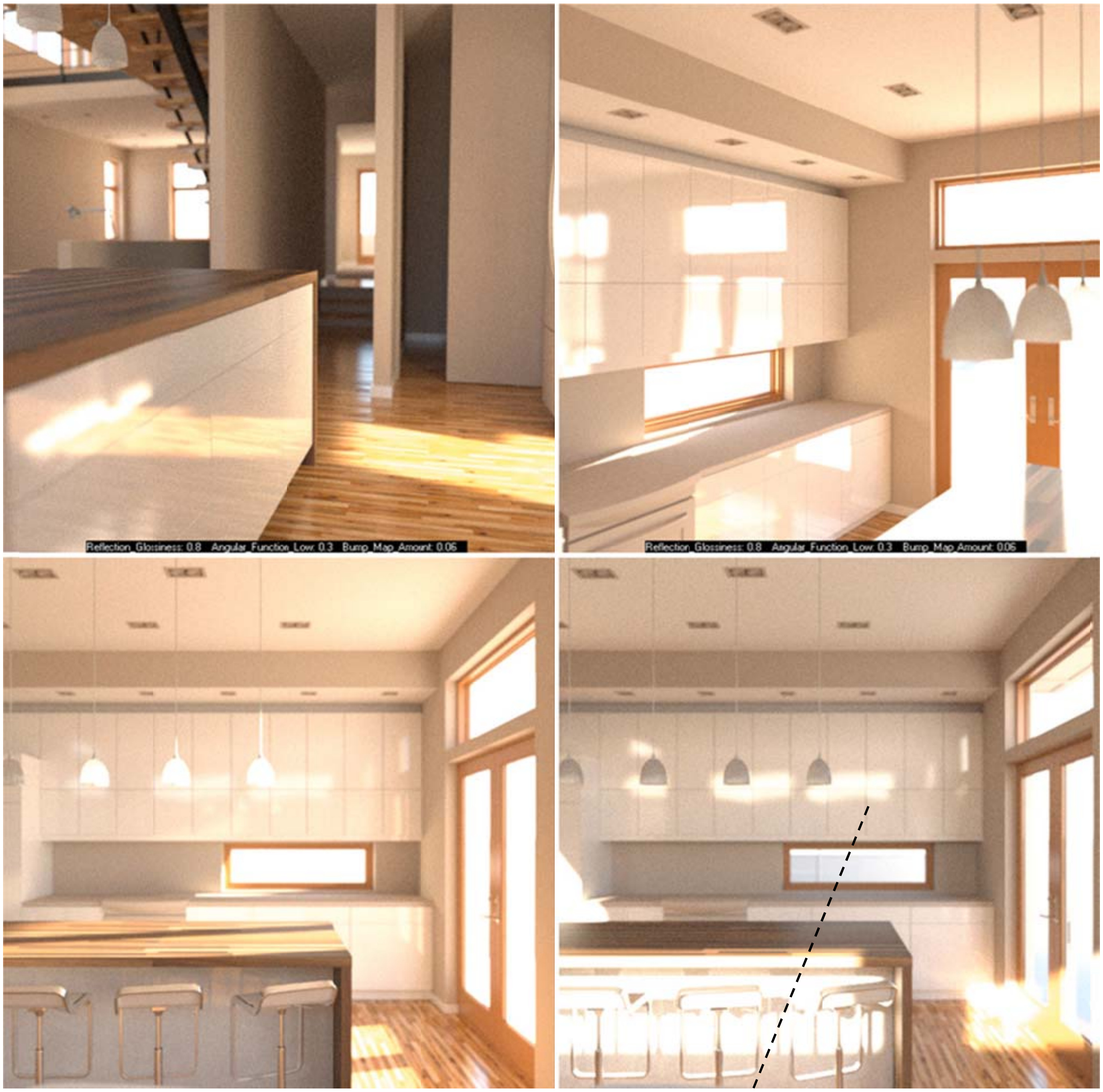


Large scale bump, very low amplitude will distort reflections slightly.

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:


Reflection_Glossiness: 0.8 Angular_Function_Low: 0.3 Bump_Map_Amount: 0.06
Reflection_Glossiness: 0.8 Angular_Function_Low: 0.3 Bump_Map_Amount: 0.06

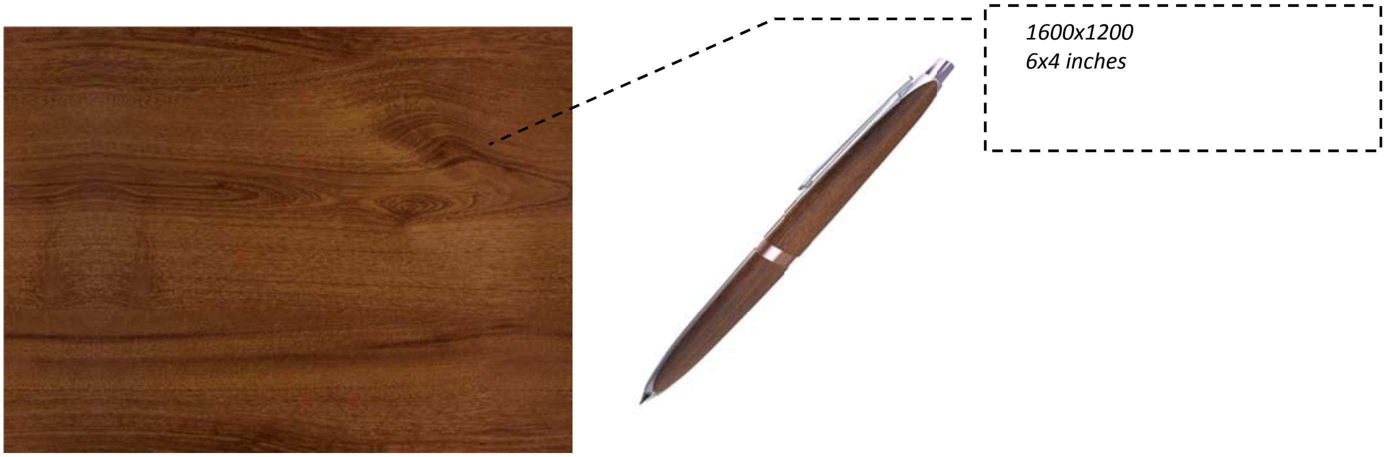
Each cabinet door and drawer has a slightly offset UVW coordinates to disjoint the large scale noise, resulting in a more natural look

Image files for materials

Different needs for different scale

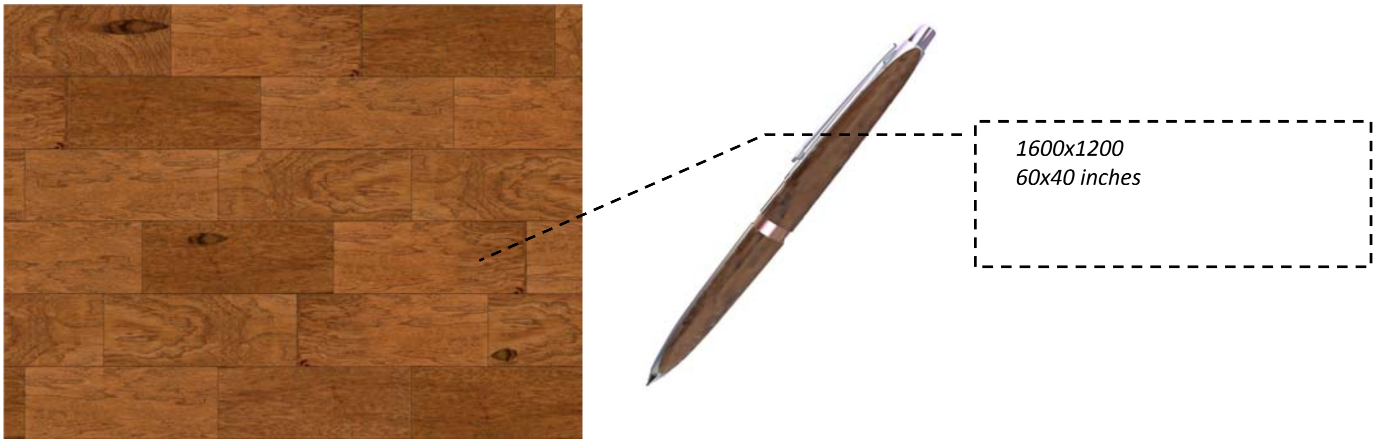
Might be worth creating two or more LOD version of some textures, depending on how you intend to use them.

Small objects textures will cover a smaller surface with high details.



1600x1200
6x4 inches

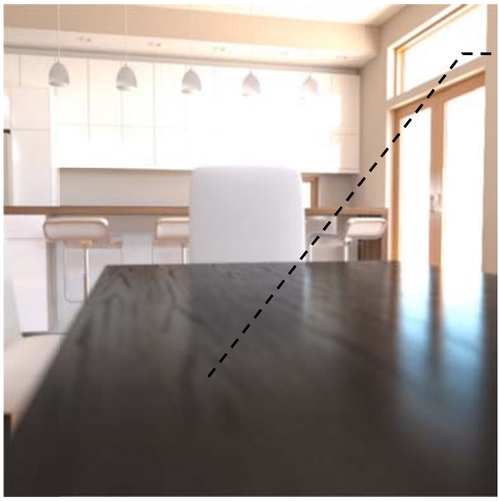
Large objects textures will cover a large surface to avoid visible tiling, but with a lower resolution.



1600x1200
60x40 inches

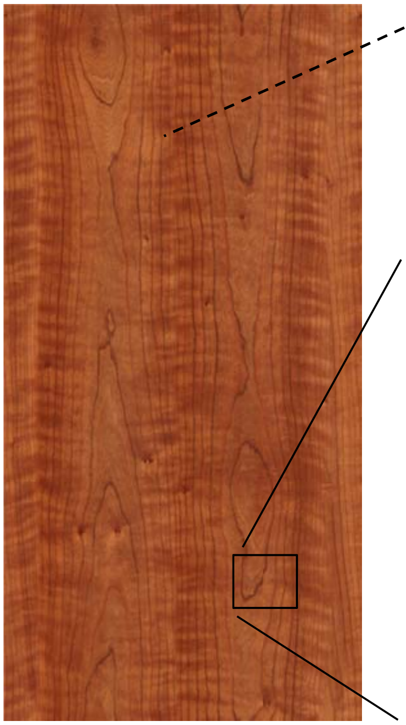
Image resolution

Elements such as furniture are often the center of attention in an image. Here are examples of wood textures that deliver high quality details in renderings:



The bump map on this material was created at a resolution of 5000 pixel wide!

Original scans, 3 boards, 2500x20000 pixels



Final texture, hand tiled 2700x5400

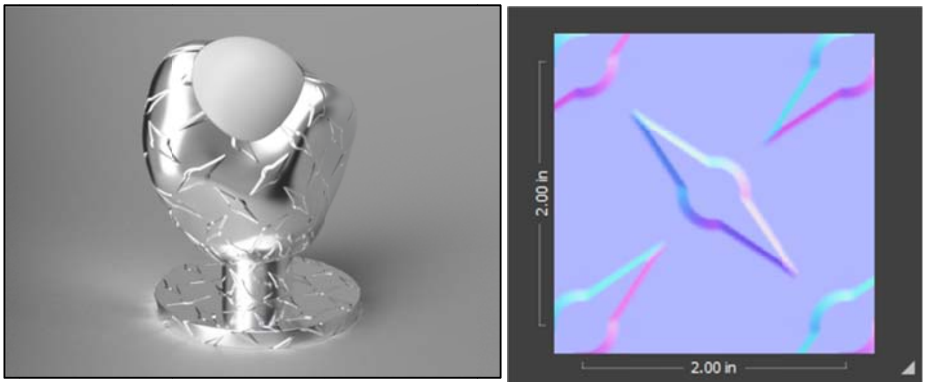
Detail

Bump and Normal Maps

Basics

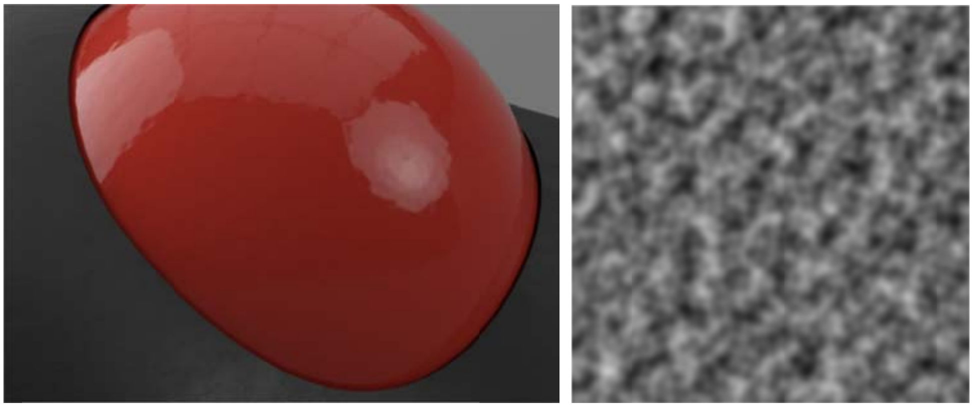
“Height maps” (Bump Maps) and “normal maps” define how surfaces are embossed or deformed instead of defining the color of the material.

“*Normal maps*” (which typically look purple/green) are generated with specialized software and cannot easily be modified in photo editing programs such as Photoshop without dedicated plugins as they define orientation vectors.



Checker plate bumps achieved with a normal map defining the embossed and flat parts of the material.

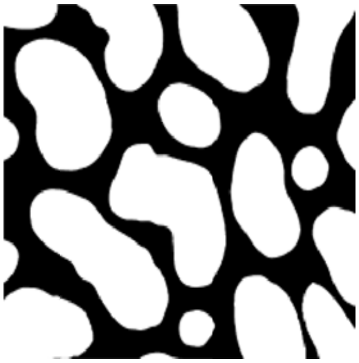
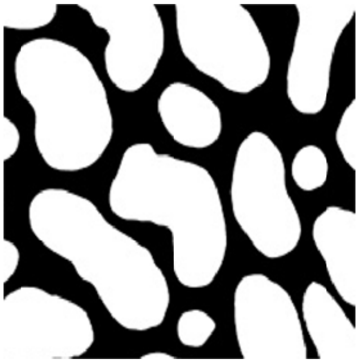
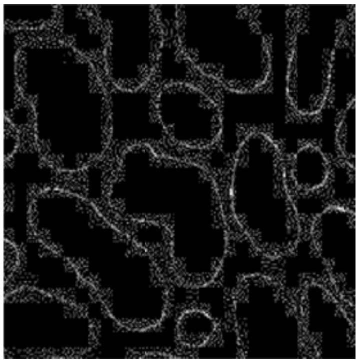
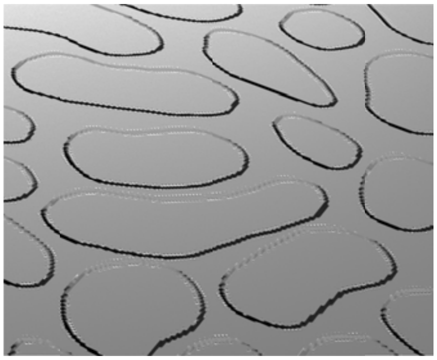
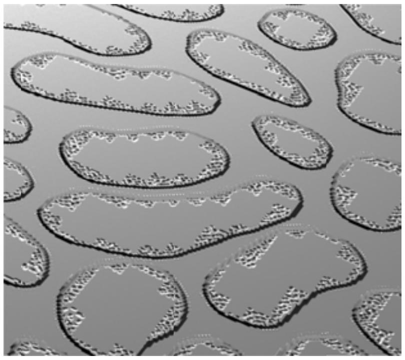
On the other hand, “*height maps*” (which are typically black and white) can be easily edited in Photoshop and are simpler to deal with as they only define elevation data.



Orange peel slightly deforming reflections achieved with a height map

File formats for Bump/Height Maps

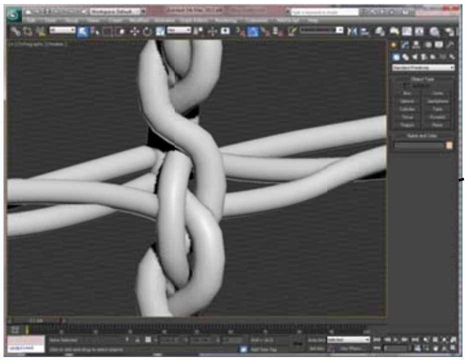
Compressed image formats can introduce artifacts that are not noticeable until rendered. JPG compression creates artifacts that produce “speckles” as shown in the example below:

PNG	JPG (Quality 90))	Difference
		
		Unnoticeable as color images, black and white images used as Height data will produce artifacts due to lossy data compression.

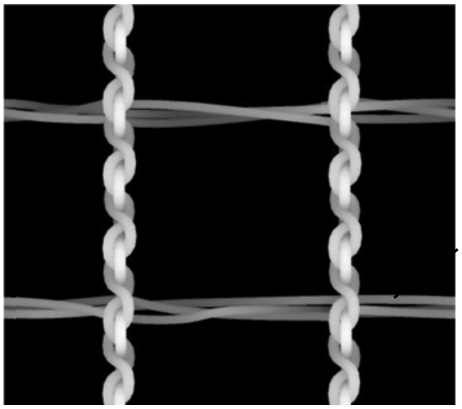
Note however, that JPG is perfectly fine for defining the color of materials.

Creating Normal Maps from 3D models

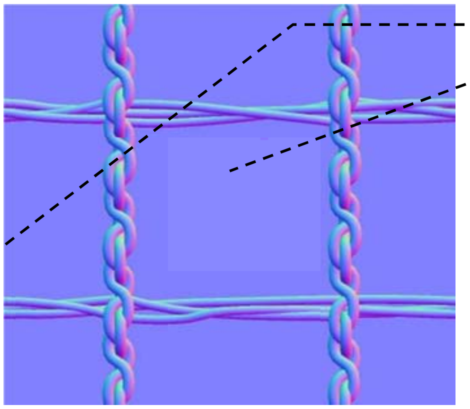
You may find useful to create a highly detailed 3d model to be used as a source to generate height and normal maps.



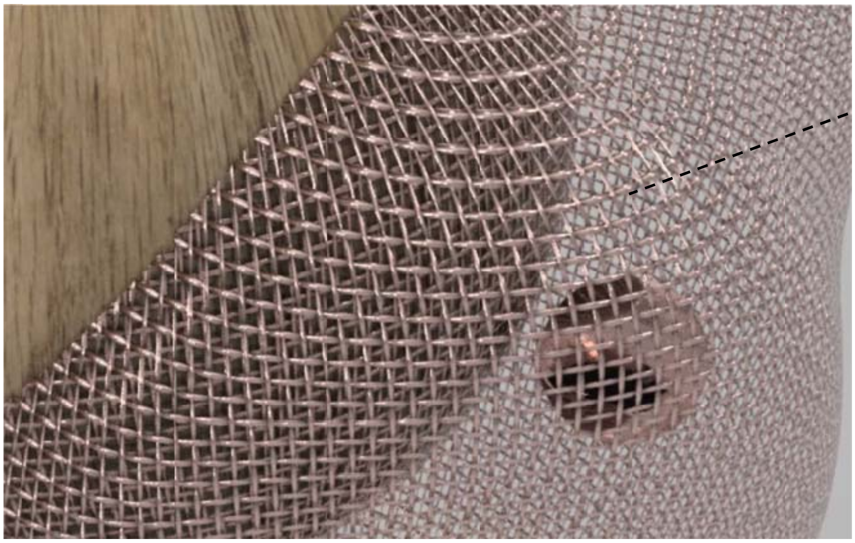
Close-up of a 3D geometry built to generate normal maps for metallic mesh



Generated Height Map



Generated Normal Map



Rendered result

Extracting normal maps (and other data) with 3ds Max

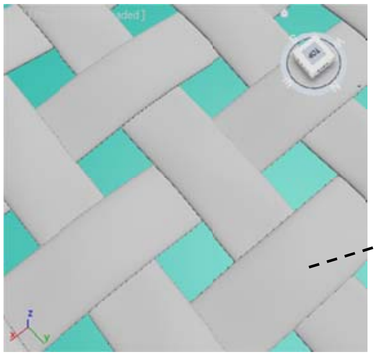
(Download the file entitled "woven 3ds max v 2016.max" in the AU download section of this class to follow along)
(Geometry created with http://www.scriptspot.com/3ds-max/mcg/mcg-simpleweave)

The idea is to project the shape of a highly detailed object on a flat plane and capture the direction and depth information into texture maps. To accomplish this, we need:

1. Detailed geometry representing the "real world" object
2. A flat plane acting as a receiving surface
3. Render to Texture to extract the information as texture files

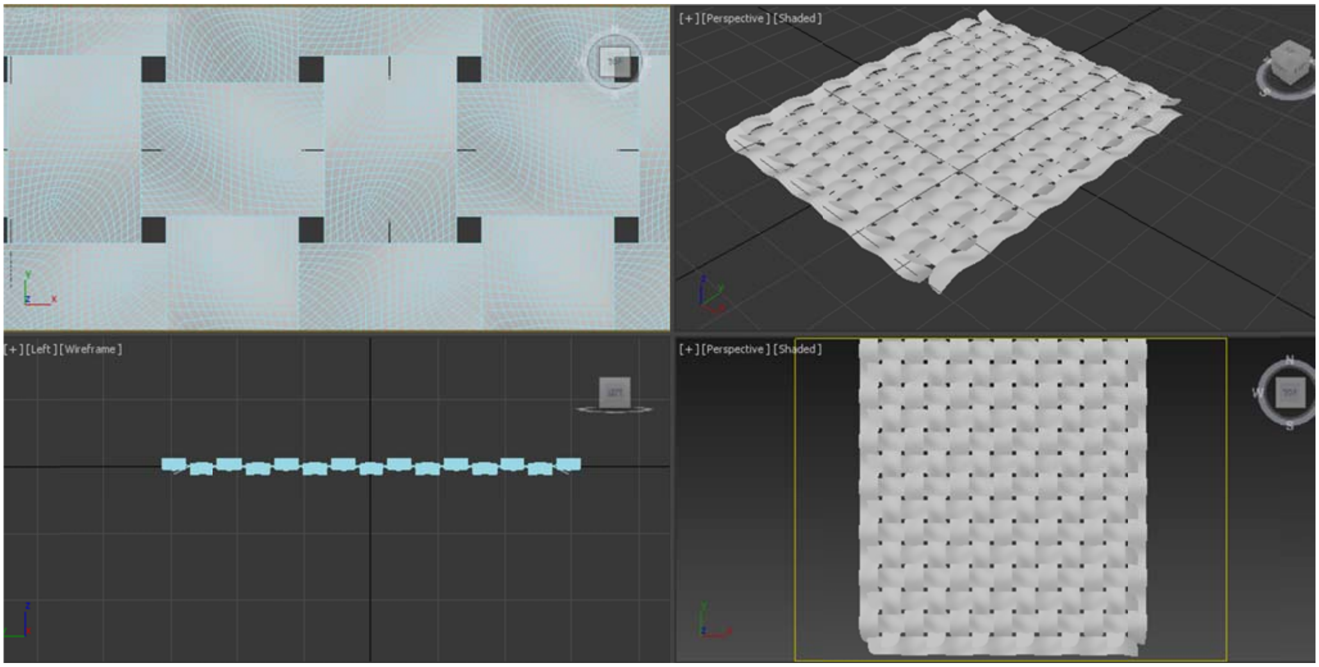
We start with a highly detailed mesh representing our finish. In this case, we modeled a woven structure meant to be applied on a chair seat. Our goal is to mimic fabric.

The surface can have holes as well.

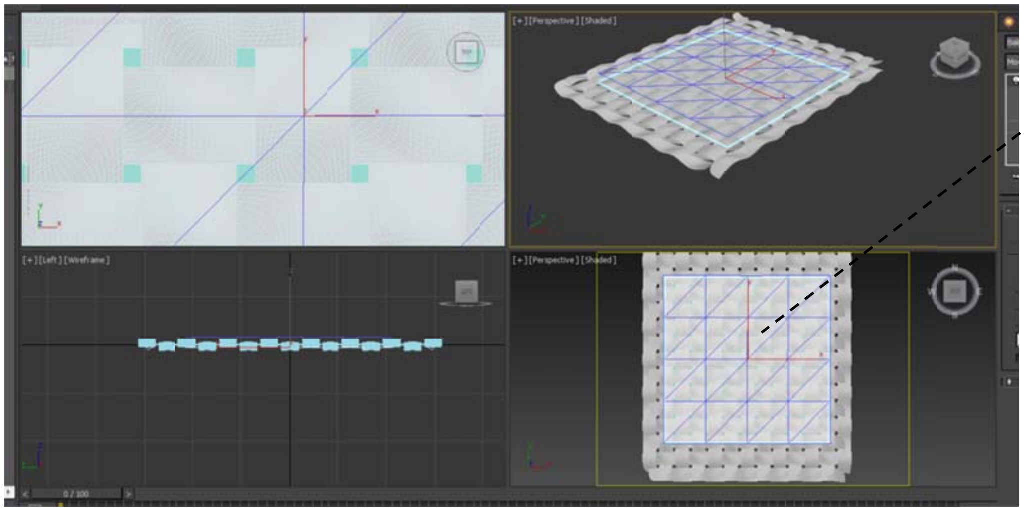


Highly detailed mesh.

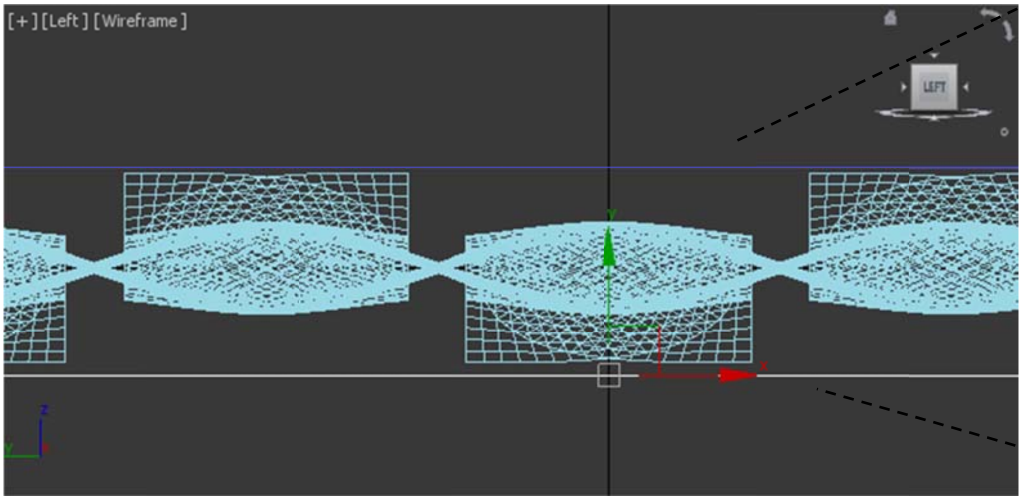
We will extract Height and Normal Maps with Render to Texture



Then, we place a “target” plane object slightly underneath our model. We apply a projection modifier on the plane and set the projection cage slightly above our mesh. (Refer to the provided model to see how it’s done)



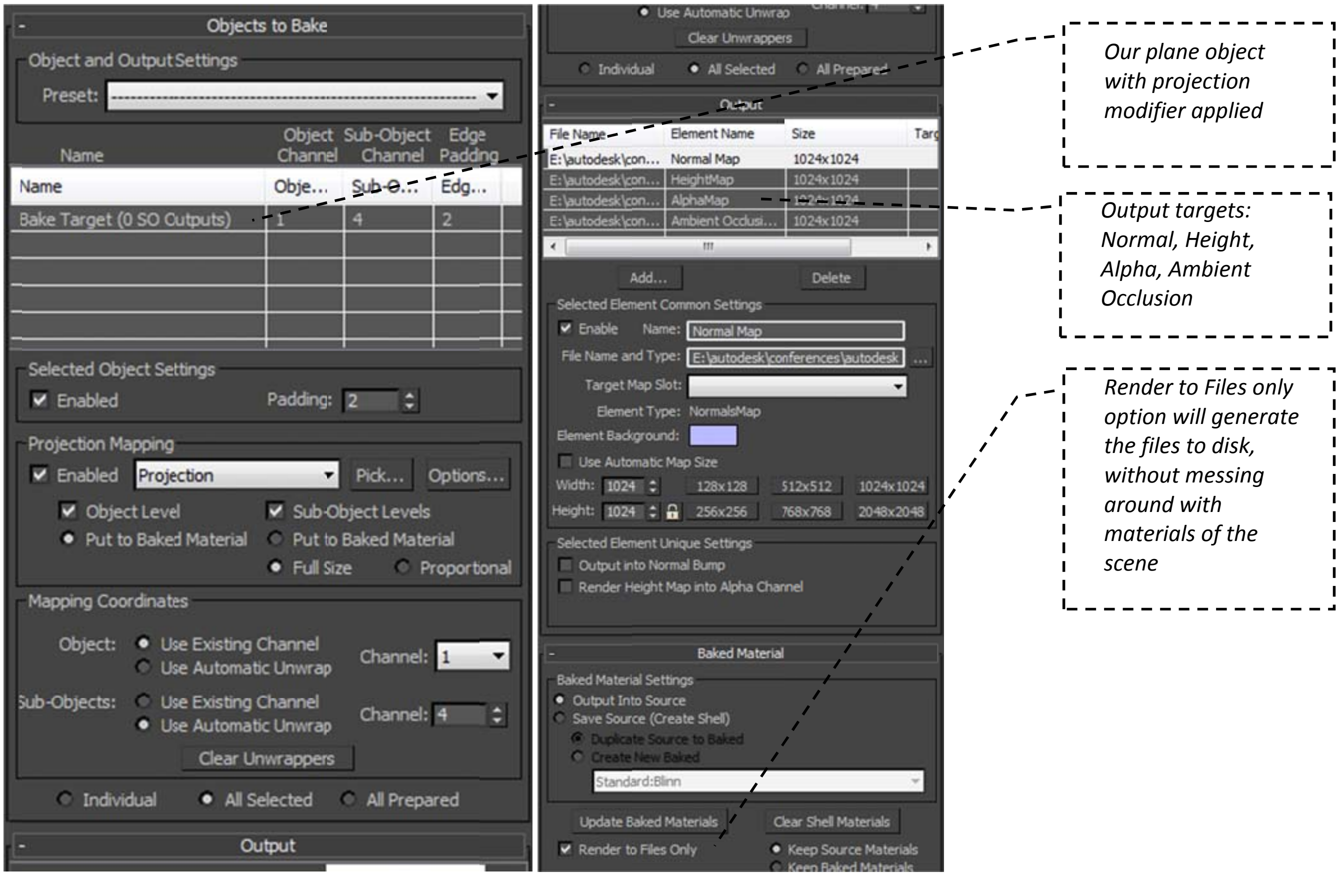
Plane object to receive textures generated



Plane Object slightly below.

Plane Object slightly below.

With Render to Texture, we then generate several images to construct our material:



Objects to Bake

Object and Output Settings

Preset: --

Name	Obje...	Sub-O...	Edg...
Bake Target (0 SO Outputs)	1	4	2

Selected Object Settings

☑ Enabled   Padding: 2

Projection Mapping

☑ Enabled   Projection   Pick...   Options...

☑ Object Level   ☑ Sub-Object Levels

● Put to Baked Material   ○ Put to Baked Material

● Full Size   ○ Proportional

Mapping Coordinates

Object: ● Use Existing Channel ○ Use Automatic Unwrap   Channel: 1

Sub-Objects: ○ Use Existing Channel ● Use Automatic Unwrap   Channel: 4

Clear Unwrappers

○ Individual   ● All Selected   ○ All Prepared

Output

● Use Automatic Unwrap

Clear Unwrappers

○ Individual   ● All Selected   ○ All Prepared

Output

File Name	Element Name	Size	Targ
E:\autodesk\con...	Normal Map	1024x1024	
E:\autodesk\con...	HeightMap	1024x1024	
E:\autodesk\con...	AlphaMap	1024x1024	
E:\autodesk\con...	Ambient Occlusi...	1024x1024	

Add...   Delete

Selected Element Common Settings

☑ Enable   Name: Normal Map

File Name and Type: E:\autodesk\conferences\autodesk ...

Target Map Slot:

Element Type: NormalsMap

Element Background:

☐ Use Automatic Map Size

Width: 1024   128x128   512x512   1024x1024

Height: 1024   256x256   768x768   2048x2048

Selected Element Unique Settings

☐ Output into Normal Bump

☐ Render Height Map into Alpha Channel

Baked Material

Baked Material Settings

● Output Into Source

○ Save Source (Create Shell)

● Duplicate Source to Baked

○ Create New Baked

Standard:Blinn

Update Baked Materials   Clear Shell Materials

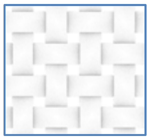
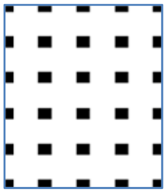
☑ Render to Files Only   ● Keep Source Materials   ○ Keep Baked Materials

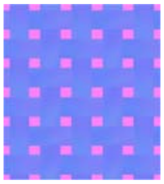
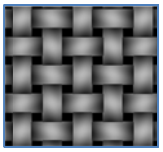
Our plane object with projection modifier applied

Output targets: Normal, Height, Alpha, Ambient Occlusion

Render to Files only option will generate the files to disk, without messing around with materials of the scene

Render to Texture Dialog

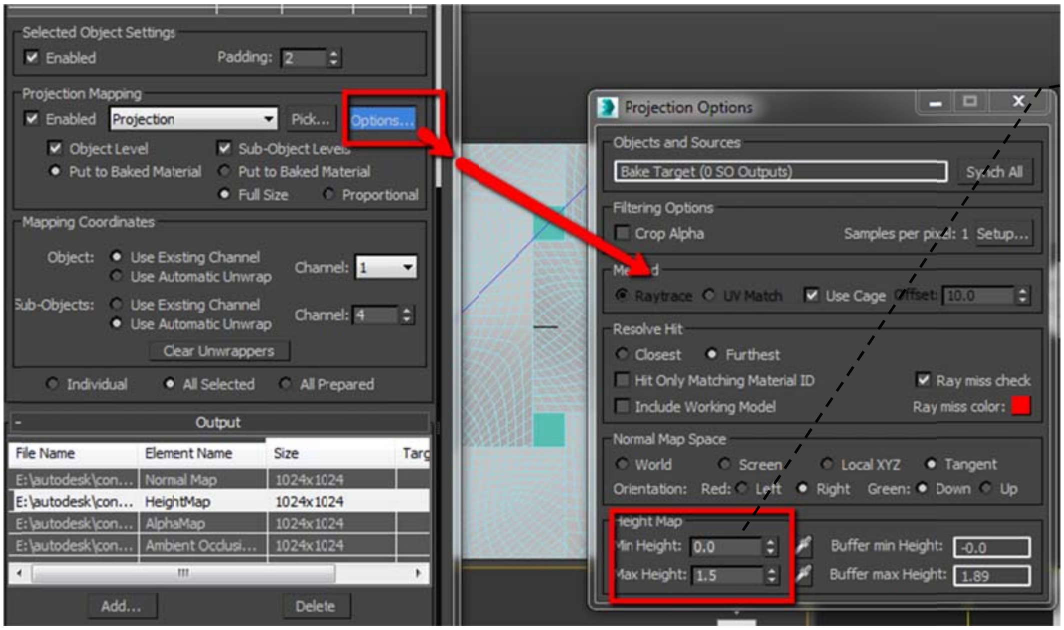
Ambient Occlusion Map (mental ray only)		Always useful to modulate the color of your materials or roughness.
Alpha		Useful to generate cutout maps (holes in the geometry)

Normal		Must be used in products supporting normal maps. 3ds Max, Fusion 360 can handle them. Revit, AutoCAD cannot. *(Note: mental ray has bugs with normal maps generation with holes / alpha. Use the Default Scanline Renderer)*
Height		Used for traditional "Bump" maps in all Autodesk products. *(See note below about setting the right range).*

Setting the correct range for Height Map extraction:

Height Maps need to be created in such a way that they contain the maximum possible range where a black pixel (0) represents the lowest part of the detailed mesh and a white pixel (1.0) represents the highest part of the detailed mesh.

This is adjusted in the Projection Mapping Options dialog:



Selected Object Settings
☑ Enabled Padding: 2

Projection Mapping
☑ Enabled Projection Pick... Options...
☑ Object Level ☑ Sub-Object Levels
● Put to Baked Material ○ Put to Baked Material
● Full Size ○ Proportional

Mapping Coordinates
Object: ● Use Exsting Channel ○ Use Automatic Unwrap Channel: 1
Sub-Objects: ○ Use Exsting Channel ● Use Automatic Unwrap Channel: 4
Clear Unwrappers
○ Individual ● All Selected ○ All Prepared

Output

File Name	Element Name	Size	Targ
E:\autodesk\con...	Normal Map	1024x1024	
E:\autodesk\con...	HeightMap	1024x1024	
E:\autodesk\con...	AlphaMap	1024x1024	
E:\autodesk\con...	Ambient Occlusi...	1024x1024	

Add... Delete

Projection Options

Objects and Sources
Bake Target (0 SO Outputs) Synch All

Filtering Options
□ Crop Alpha Samples per pixel: 1 Setup...

Method
● Raytrace ○ UV Match ☑ Use Cage Offset: 10.0

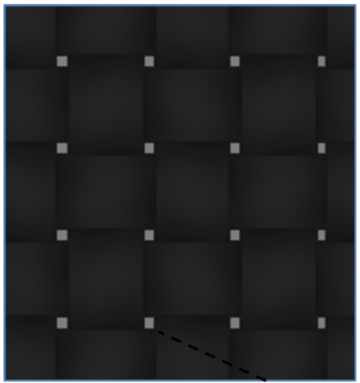
Resolve Hit
○ Closest ● Furthest
□ Hit Only Matching Material ID ☑ Ray miss check
□ Include Working Model Ray miss color: 🟥

Normal Map Space
○ World ○ Screen ○ Local XYZ ● Tangent
Orientation: Red: ○ Left ● Right Green: ● Down ○ Up

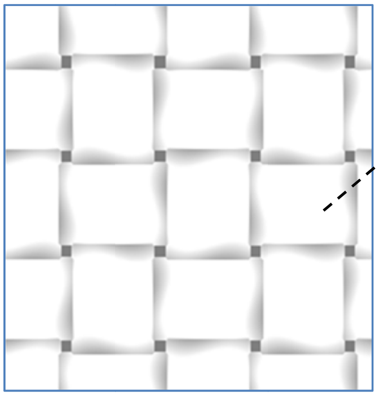
Height Map
Min Height: 0.0 Buffer min Height: -0.0
Max Height: 1.5 Buffer max Height: 1.89

Height Map range must be double checked.

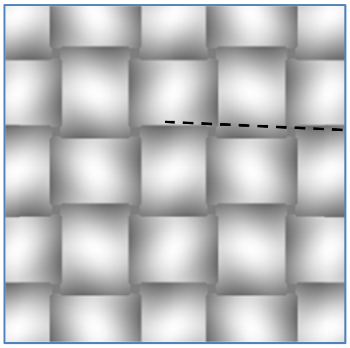
In our example, height maps that could look like these images would be incorrect:



Too dark!



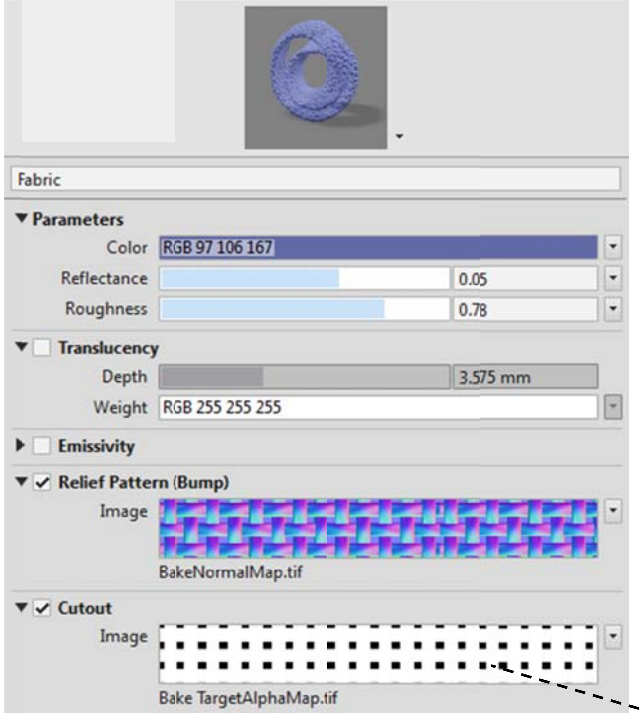
Missing highest mesh data, resulting in pixel clipping.



Correct Range

Using the texture maps

In the Material Editor, you simply use the resulting images as follow:



Fabric

▼ Parameters

Color: RGB 97 106 167

Reflectance: 0.05

Roughness: 0.78

▼ □ Translucency

Depth: 3.575 mm

Weight: RGB 255 255 255

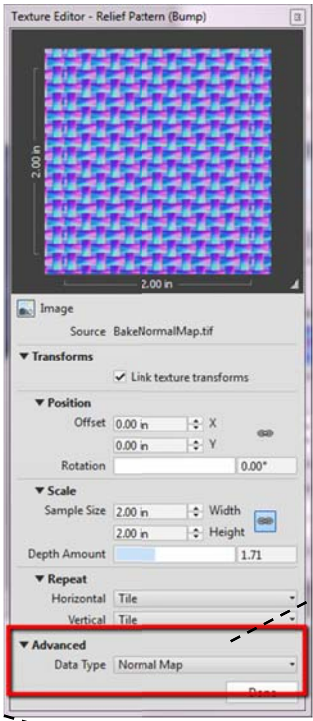
▶ □ Emissivity

▼ ☑ Relief Pattern (Bump)

Image: BakeNormalMap.tif

▼ ☑ Cutout

Image: Bake TargetAlphaMap.tif



Texture Editor - Relief Pattern (Bump)

Image

Source: BakeNormalMap.tif

▼ Transforms

☑ Link texture transforms

▼ Position

Offset: 0.00 in X

0.00 in Y

Rotation: 0.00°

▼ Scale

Sample Size: 2.00 in Width

2.00 in Height

Depth Amount: 1.71

▼ Repeat

Horizontal: Tile

Vertical: Tile

▼ Advanced

Data Type: Normal Map

Normal Map

In Fusion 360: Make sure you set the Data Type option to "Normal Map"

In Revit or AutoCAD, use the Height Map instead

Alpha Map

Rendered in Fusion 360 , using the generated textures