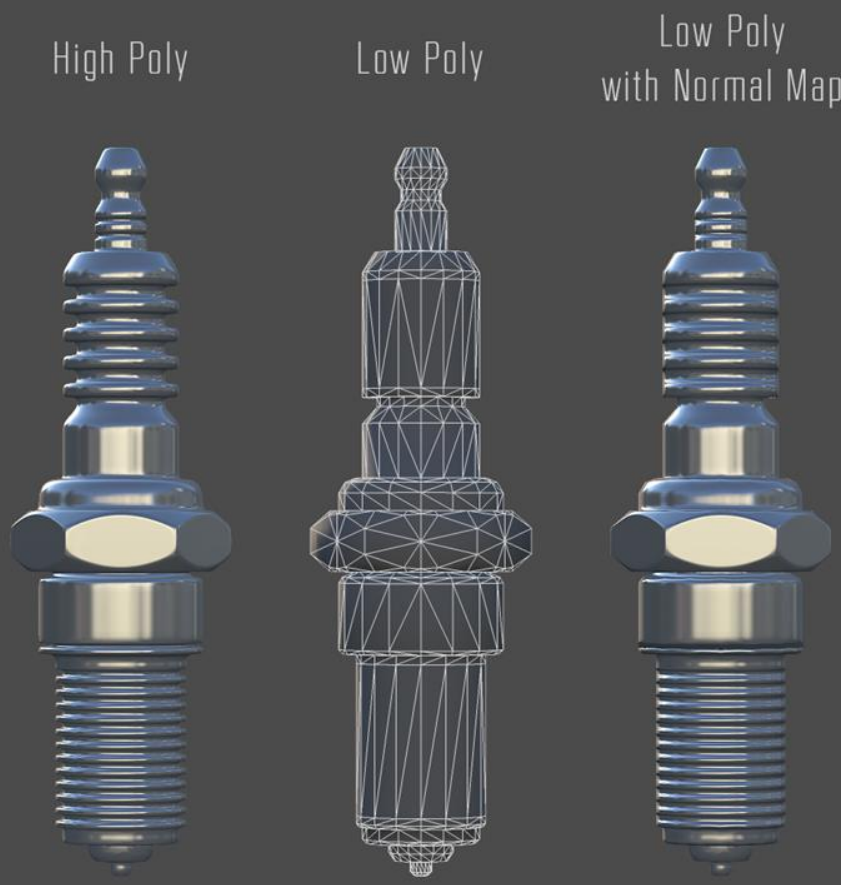
Normal Mapping For Games & Realtime Content

Introduction

WHAT IS A NORMAL MAP?

Normal Maps 101

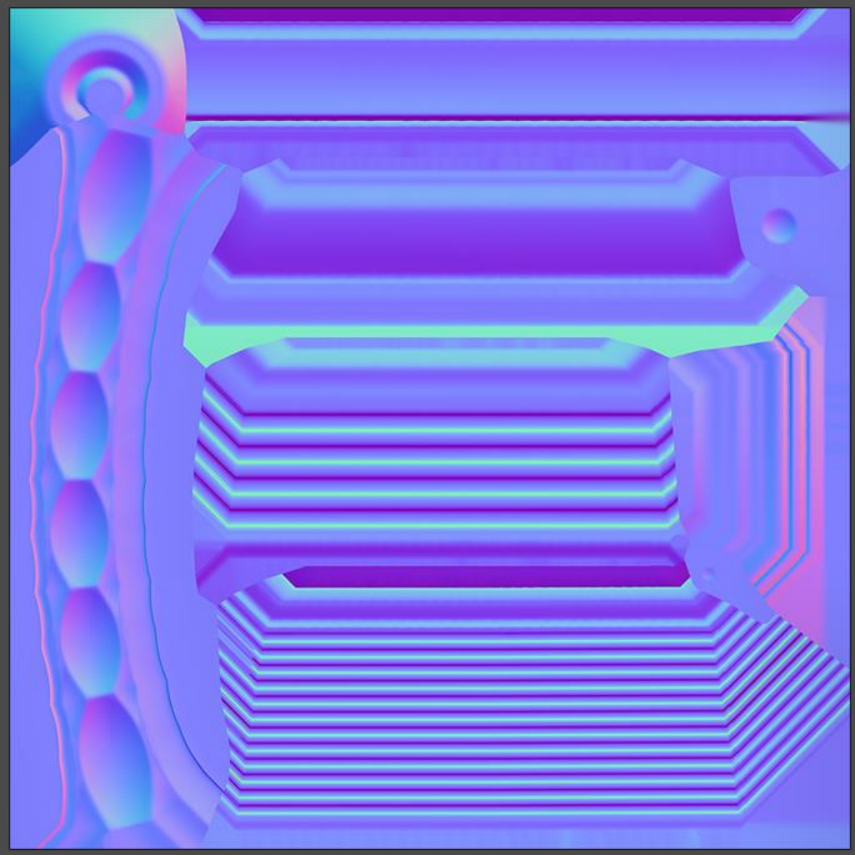
-Used to fake the detail of a high poly mesh
-Each pixel found in the normal map has an RGB value
-Each RGB value represents a normal
-Each normal represents a vector
-A vector is a value that represents the surface slope of the high poly mesh.
-Different RGB values represent different directions
-These values control the direction that the pixel is facing in 3d space
-Not like other textures (Diffuse, Specular, Gloss etc.)
-Essentially encoded data stored within a texture
-Allows for very detailed rendering for minimal cost
-Doesn't alter the silhouette of the low poly mesh



TANGENT SPACE vs OBJECT SPACE vs DERIVATIVE

Tangent Space

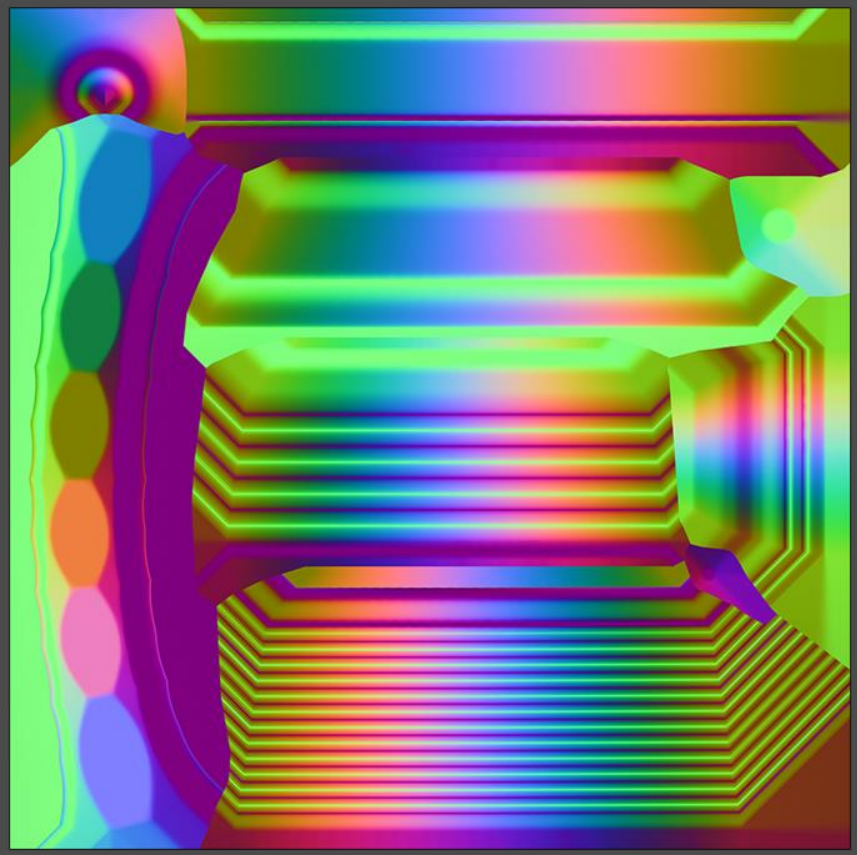
- Cheap in memory versus OS
- Compresses well versus OS
- Only red & green channels are used (blue is recreated in the shader).
- Compression algorithms are getting better (Crytek's modified 3Dc etc.)
- A good encoder makes all the difference!
- Not always mesh specific
- Tiling & mirroring are trivial
- Easy to overlay detail maps
- Deformation is trivial
- Smooths badly versus OS
- No standard Tangent basis (mikktspace is the most popular) :)



TANGENT SPACE vs OBJECT SPACE vs DERIVATIVE

Object Space

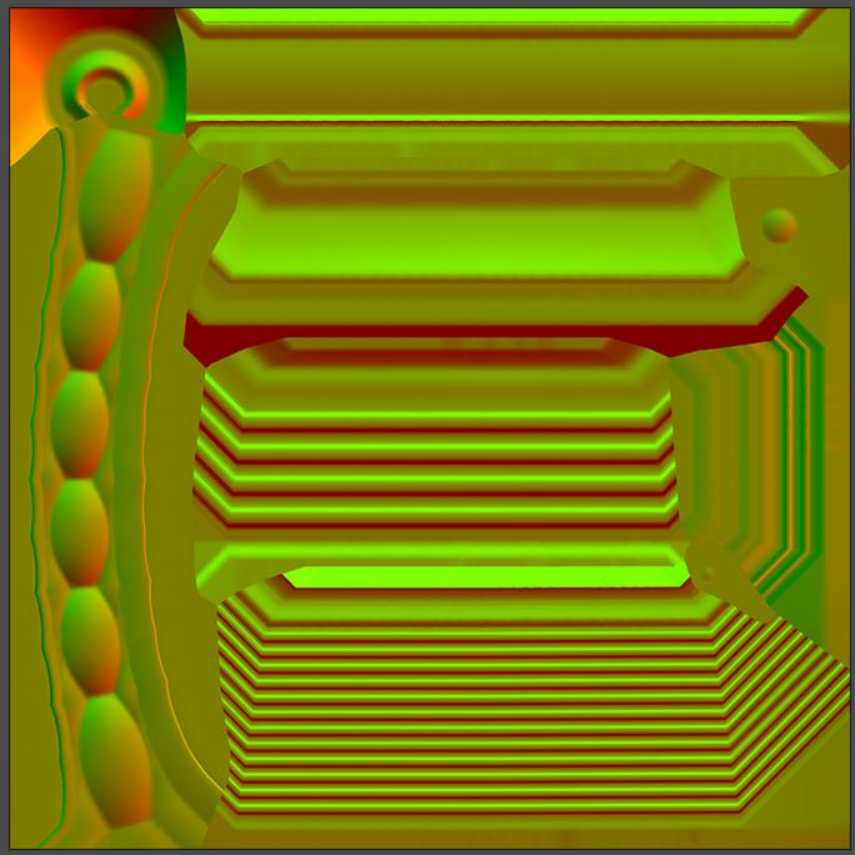
- Smooths very well versus TS
- Great for texturing (selection masks)
- Works in any application
- Expensive in memory versus TS
- Compresses poorly versus TS
- All channels are used
(blue cannot be recreated in the shader)
- Mesh specific & hard to reuse
- Tiling & mirroring are non-trivial
- Hard to overlay detail maps
- XYZ may not be baked in the correct order
for your application/engine (fix in Photoshop)



TANGENT SPACE vs OBJECT SPACE vs DERIVATIVE

Derivative

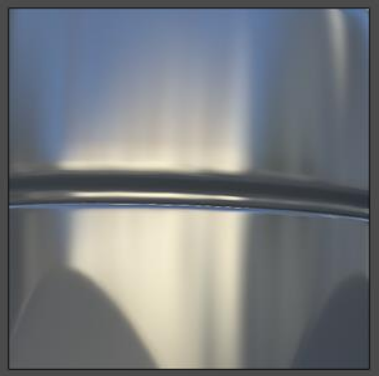
- Cheap in memory versus OS & TS
- Compresses well versus OS
- Only red & green channels are used
- Doesn't require per vertex normals (yay!)
- No need to precompute tanget space
- Not always mesh specific
- Works well on tessellated surfaces
- Tiling & mirroring are trivial
- Doesn't require normalization
- Easy to overlay detail maps (easier than TS)
- Doesn't require per vertex normals
- Viewing supported in Blender
- Baking supported in xNormal
- Smooths badly versus OS & TS when using course geometry.
- Shading is more reliant on the shading of a mesh (higher resolution geometry reduces these issues)

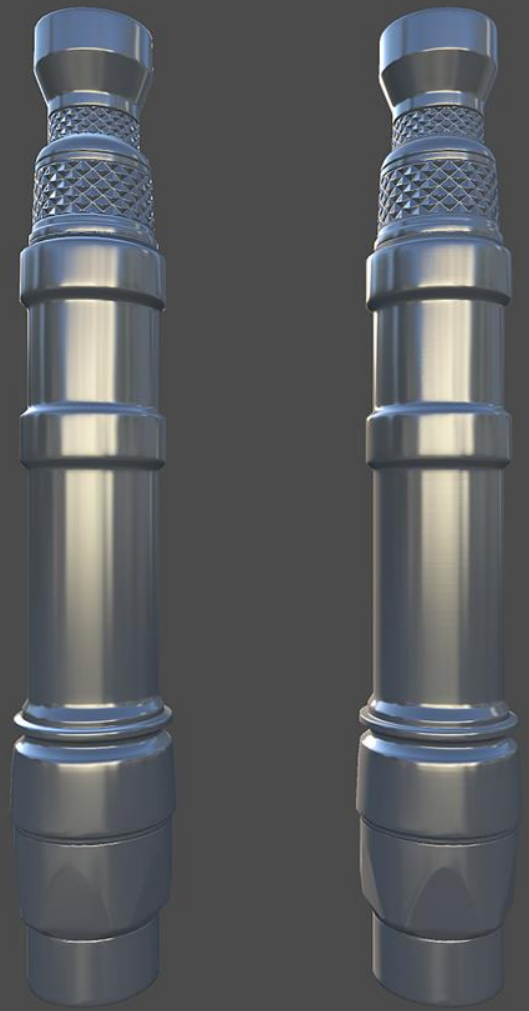


BIT DEPTH - 16-BITS/CHANNEL VERSUS 8-BITS/CHANNEL

16-bits/channel

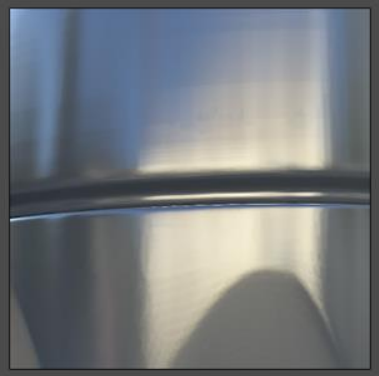
- -65536 Maximum unique values
- -High precision
- -More values equals greater accuracy
- -Good quality compression (3Dc)
- -Expensive (relatively) to store
- -Large Files: 24MB for 2048^2 when uncompressed





8-bits/channel

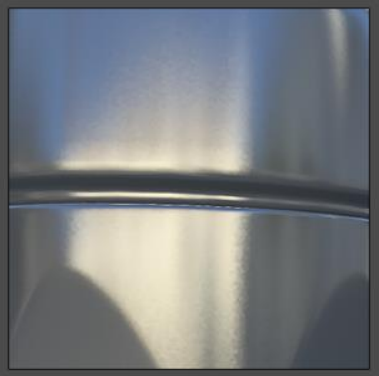
- -256 Maximum unique values
- -Low precision
- -Poor accuracy (banding)
- -Std. quality compression (DXT1)
- -Cheap to store
- -Smaller files: 2MB for 2048^2 when uncompressed



BIT DEPTH - 16-BITS/CHANNEL VERSUS 8-BITS/CHANNEL

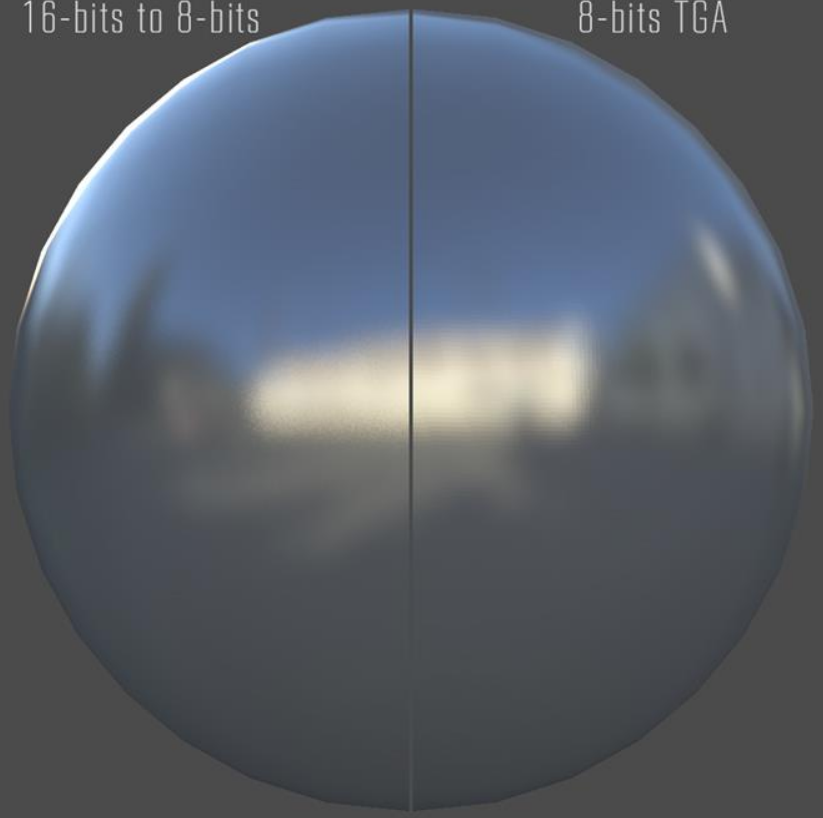
16-bits/channel

-Work in 16-bits/channel
-Convert to 8-bits/channel on export
-Now has 256 Maximum unique values
-Dithering ftw?
-Still better quality than natively baking at 8-bits/channel!



Dithered
16-bits to 8-bits

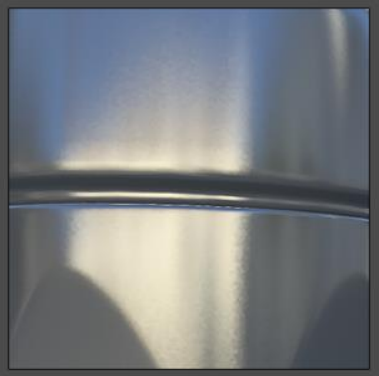
Uncompressed
8-bits TGA



BIT DEPTH - 16-BITS/CHANNEL VERSUS 8-BITS/CHANNEL

16-bits/channel

-Work in 16-bits/channel
-Convert to 8-bits/channel on export
-Now has 256 Maximum unique values
~~-Dithering ftw?~~
-Use a good quality encoder!
-Still better quality than natively baking at 8-bits/channel!



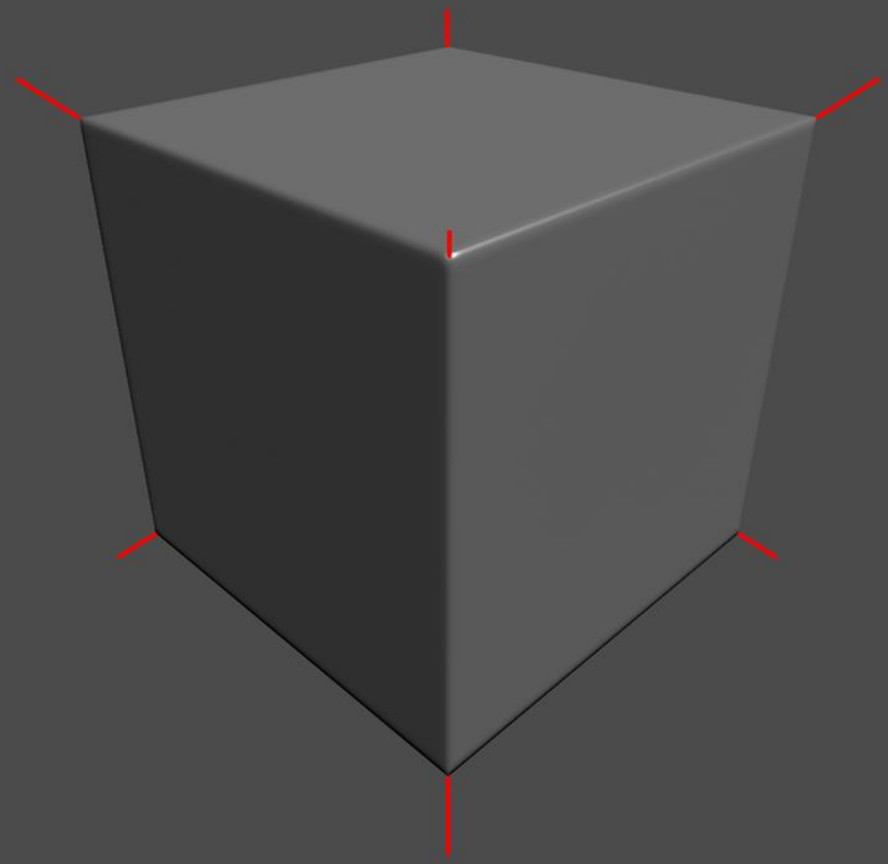
Crytek Modified 3Dc
16-bits to 8-bits

Uncompressed
8-bits TGA

PROJECTION CAGES VS UNIFORM RAY DISTANCE

Uniform ray distance

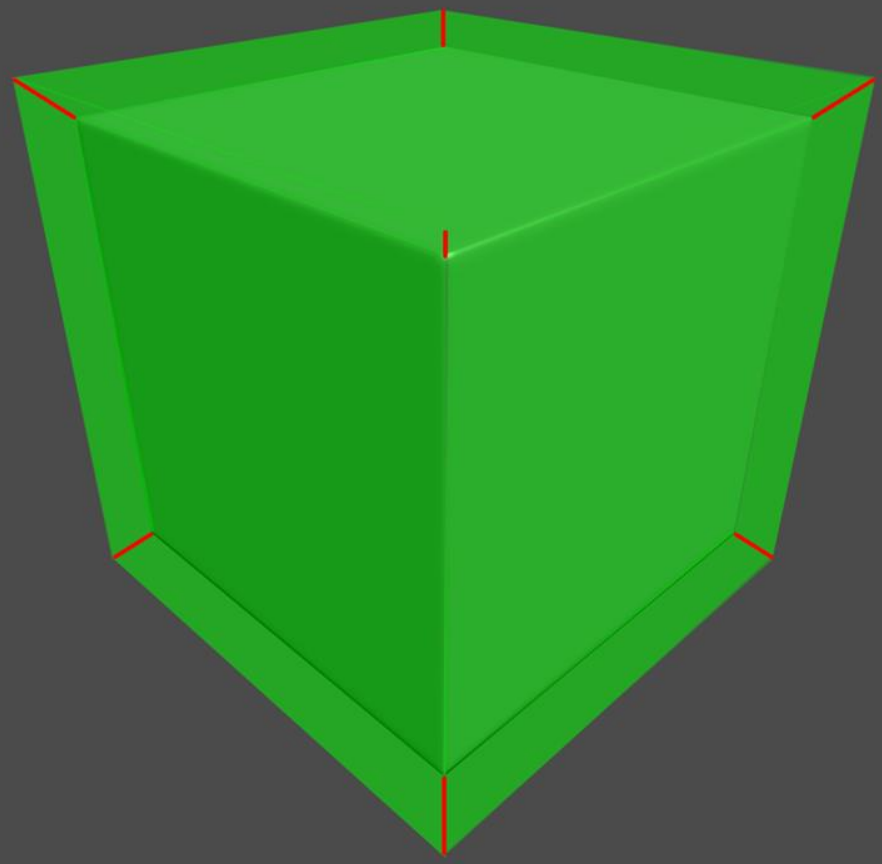
-The most basic way to bake Normal maps
-Most tools support this method of baking
-Rays are cast outwards along each vertex normal
-At a set distance the rays is cast back inwards towards the mesh
-Normals are sampled where the rays intersect with the highpoly mesh



PROJECTION CAGES VS UNIFORM RAY DISTANCE

Uniform ray distance

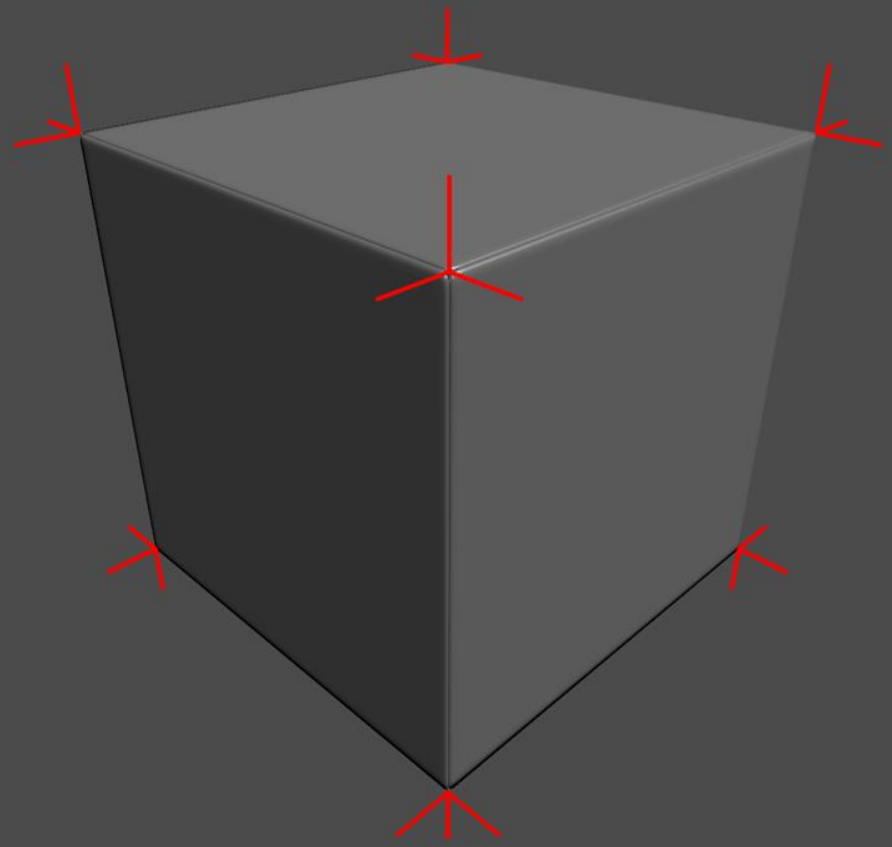
-The most basic way to bake Normal maps
-Most tools support this method of baking
-Rays are cast outwards along each vertex normal
-At a set distance the rays is cast back inwards towards the mesh
-Normals are sampled where the rays intersect with the highpoly mesh
-Works ok on planes or all soft edges as all areas are covered



PROJECTION CAGES VS UNIFORM RAY DISTANCE

Uniform ray distance

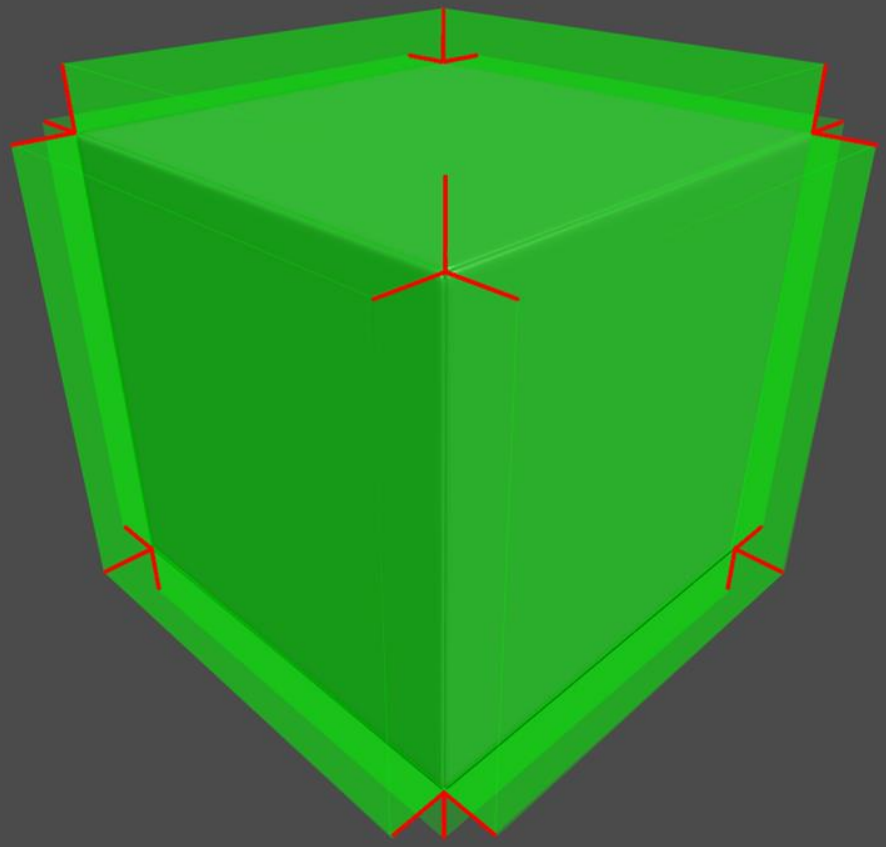
-The most basic way to bake Normal maps
-Most tools support this method of baking
-Rays are cast outwards along each vertex normal
-At a set distance the rays is cast back inwards towards the mesh
-Normals are sampled where the rays intersect with the highpoly mesh
-Works ok on planes or all soft edges as all areas are covered
-Spit vertex normals cause errors & seams in the Normal map



PROJECTION CAGES VS UNIFORM RAY DISTANCE

Uniform ray distance

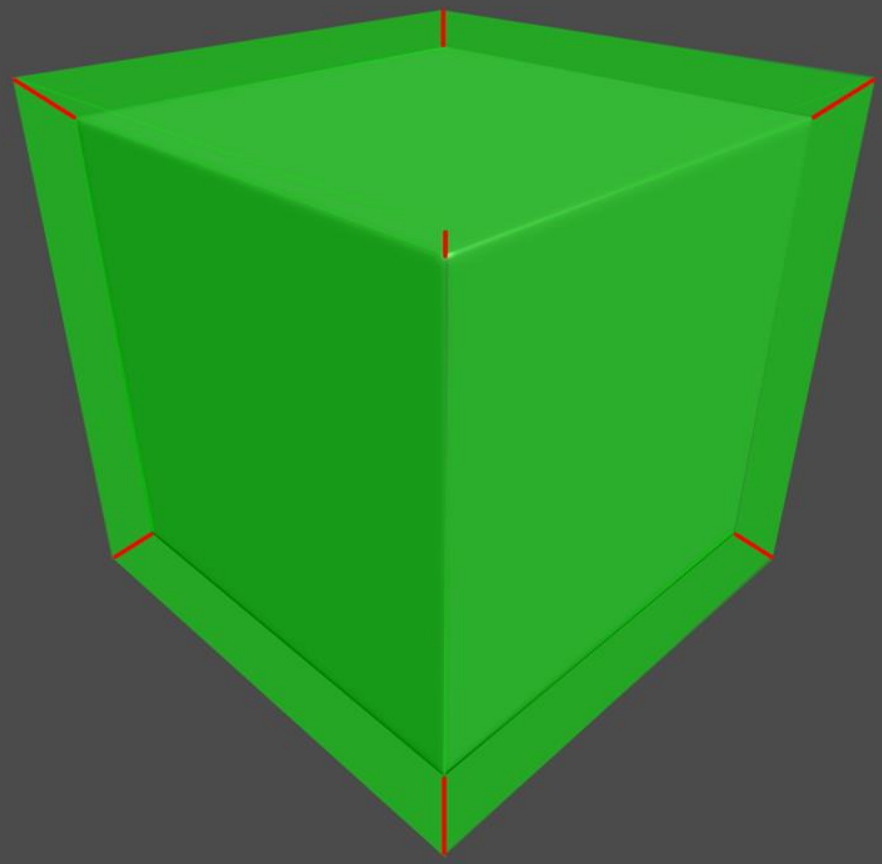
-The most basic way to bake Normal maps
-Most tools support this method of baking
-Rays are cast outwards along each vertex normal
-At a set distance the rays is cast back inwards towards the mesh
-Normals are sampled where the rays intersect with the highpoly mesh
-Works ok on planes or all soft edges as all areas are covered
-Spit vertex normals cause errors & seams in the Normal map
-This is because parts of the bake are missed



PROJECTION CAGES VS UNIFORM RAY DISTANCE

Projection Cages

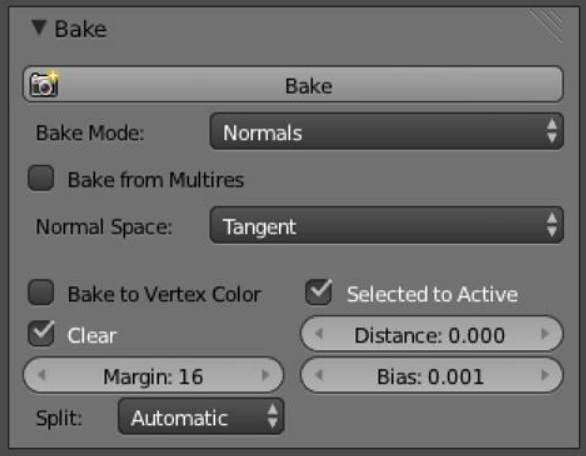
-Projection cages don't suffer from the same problems
-Cages are an inflated duplicate of the low poly mesh
-Vertex normal in the cage are averaged so no missed areas!
-Rays are cast inwards from the cage
-Normals are sampled where the rays intersect with the highpoly mesh
-Great for all meshes!
-Especially good for hands and situations where geometry is very close together
-Allows for multiple ray distances when baking
-A superior way to bake



BAKING IN BLENDER

Pros & Cons

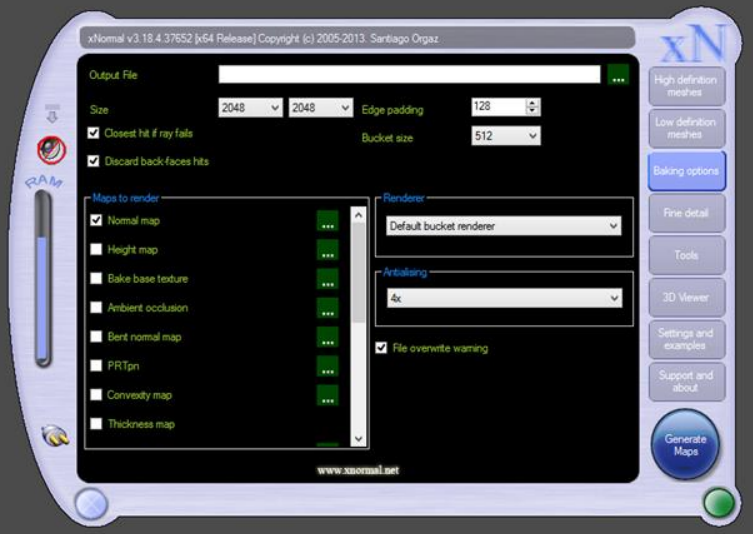
- Baking in Blender is very simple & intuitive
- Multiple normal spaces are available (Tangent, Object, World & Camera space)
- Uses Mikktspace
- Multi-res baking available
- Bakes with correct background colour
- Decent margin/bleed value by default
- No Cages (yet)
- Distance based ray cast
- Cant bake with hard edges/per vertex normals
- No axis swizzle (yet)
- Tangent & other normal spaces use different XYZ configurations
- Average quality bakes with reasonable detail preservation
- No anti-aliasing
- Color space display issues cause confusion to inexperienced users (linear vs sRGB)



BAKING IN xNormal

Pros & Cons

- Tangent & object spaces are available
- Uses Mikktspace
- Bakes with correct background colour
- Cages, distance based ray cast & blockers
- High quality bakes with excellent detail preservation
- GPU baking via CUDA & OpenRL
- Anti-aliasing
- Very powerful and configurable
- Baking in xNormal has a steep learning curve
- Poor interface
- No multi-res baking

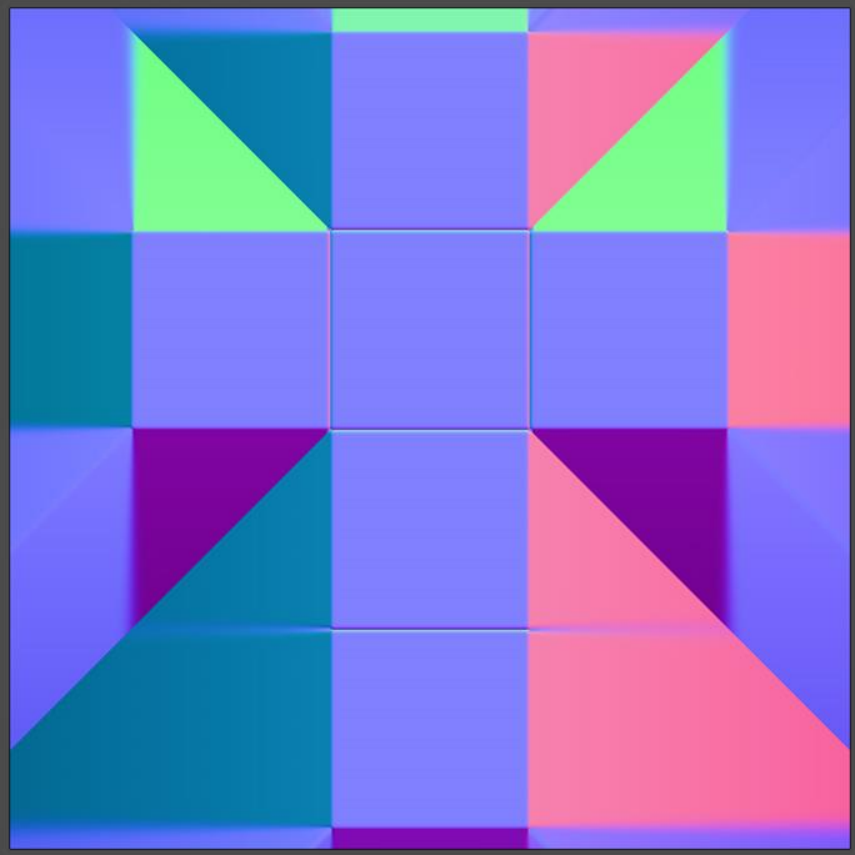


Best Practices

SPLITTING UVS

Splitting UVs

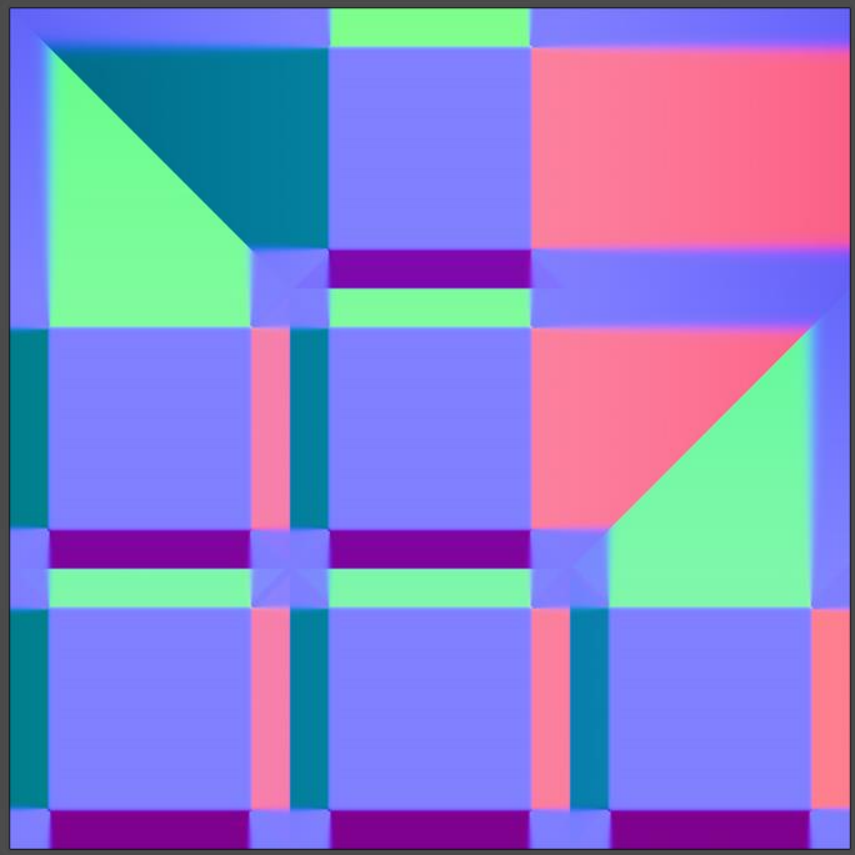
-When using hard edges you should split your UVs
-If this not done the vertices in UV space are overlaid and an average value (colour) is written
-This value is incorrect in tangent space and causes a seam
-A space of two pixels in UV space wherever there is a hard edge in 3d space is required



SPLITTING UVS

Splitting UVs

-When using hard edges you should split your UVs
-If this not done the vertices in UV space are overlaid and an average value (colour) is written
-This value is incorrect in tangent space and causes a seam
-A space of two pixels in UV space wherever there is a hard edge in 3d space is required
-This allows the normals to resolve correctly when baked
-The verts no longer share the same location in UV space
-A correct value (colour) can now be baked
-Best to unwrap first then set hard edges to UVs
-Doesn't cost any extra as extra vertices have already been created in UV space



BEST PRACTICES

The Importance Of Triangulation

-Always triangulate before baking!
-A Normal map can only compensate for the geometry present during the bake
-Incorrect geometry will cause shading errors
-Game engines may triangulate your mesh differently than your baker
-Geometry is automatically triangulated before baking
-Each quad can be triangulated in two different ways
-50% of all quads will be triangulated incorrectly
-Ngons are exponentially worse as sides increase

BEST PRACTICES

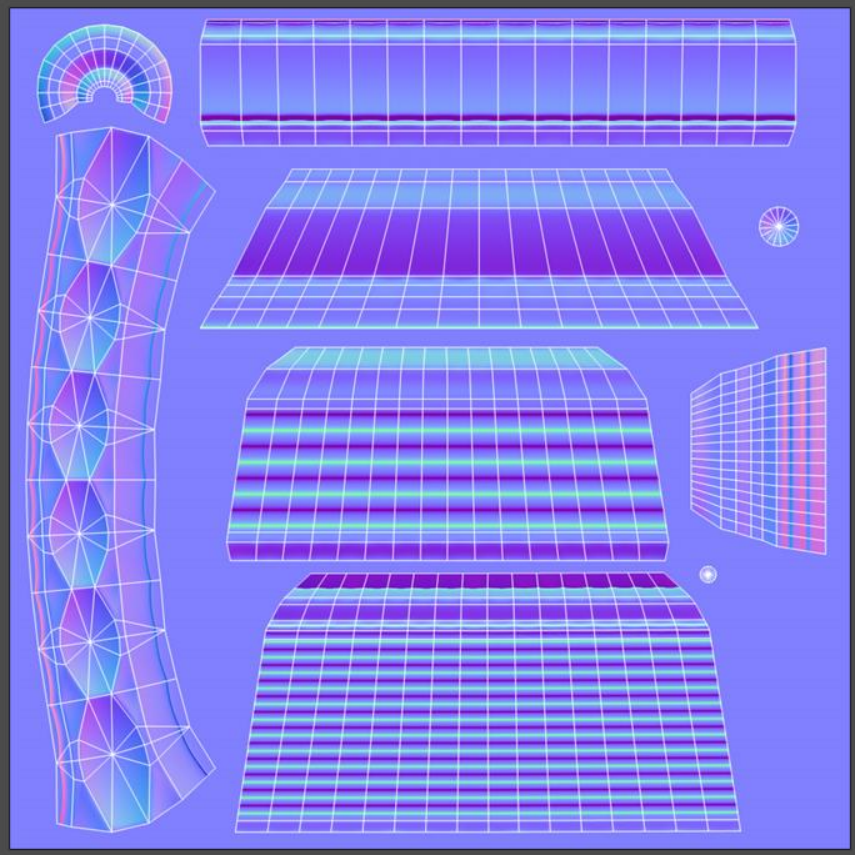
The Importance Of Triangulation

-Always triangulate before baking!
-A Normal map can only compensate for the geometry present during the bake
-Incorrect geometry will cause shading errors
-Game engines may triangulate your mesh differently than your baker
-Geometry is automatically triangulated before baking
-Each quad can be triangulated in two different ways
-50% of all quads will be triangulated incorrectly
-Ngons are exponentially worse as sides increase
-Triangulating before baking ensures no room for error
-Always use the same geometry in engine as was used when baking

BEST PRACTICES

Edge Padding/Bleed

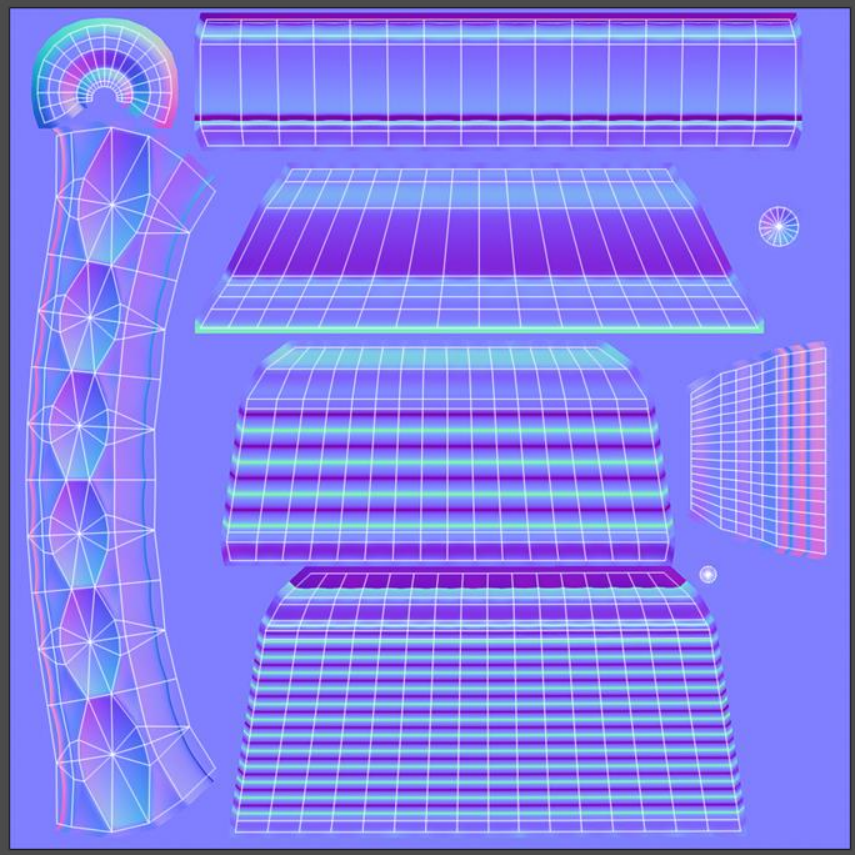
-Edge padding should be added to all bakes
-Stops the bleed between pixels when mipped
-Mipping is the automatic reduction of texture size based on object distance
-Aim for a 1px padding around each UV island at lowest mip level



BEST PRACTICES

Edge Padding/Bleed

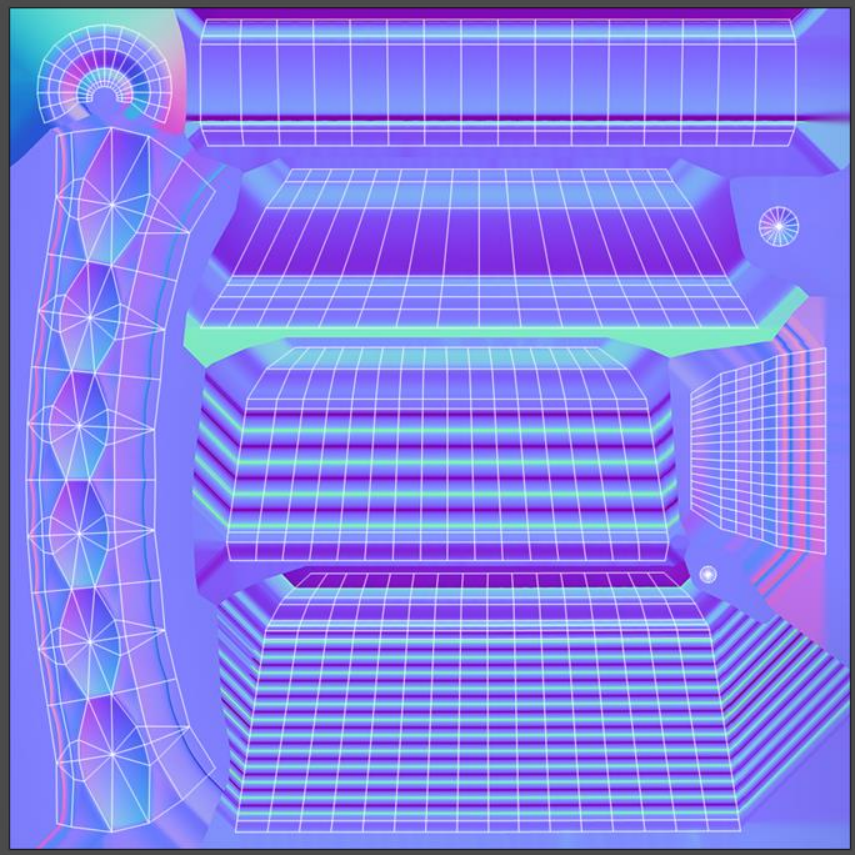
-16px of padding around each UV island should be sufficient in most cases
-This means 16px of padding per island & not between each island
-16px of padding gives 5 mip levels (16/8/4/2/1)



BEST PRACTICES

Edge Padding/Bleed

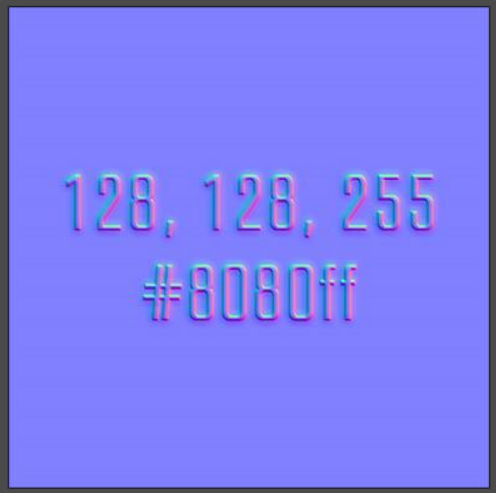
-Better still is 100% padding to the border of the texture
-You can perform 100% padding as a final step
-Leaves no margin for error
-How it looks on the texture is irrelevant
-How it looks in-game is most important



BEST PRACTICES

Background Colour

-The background colour of a Normal map is important!
-Not all colours represent valid normals
-Never use black as it represents a downward facing normal
-Tangent space Normal maps don't have downward facing normals
-In sRGB the correct value should be 127.5, 127.5, 255 but this isn't possible.
-Instead we use 128, 128, 255 (#8080FF) or .5, .5, 1.0 in linear space
-128, 128, 255 (#8080FF) represents a flat normal (no surface change)



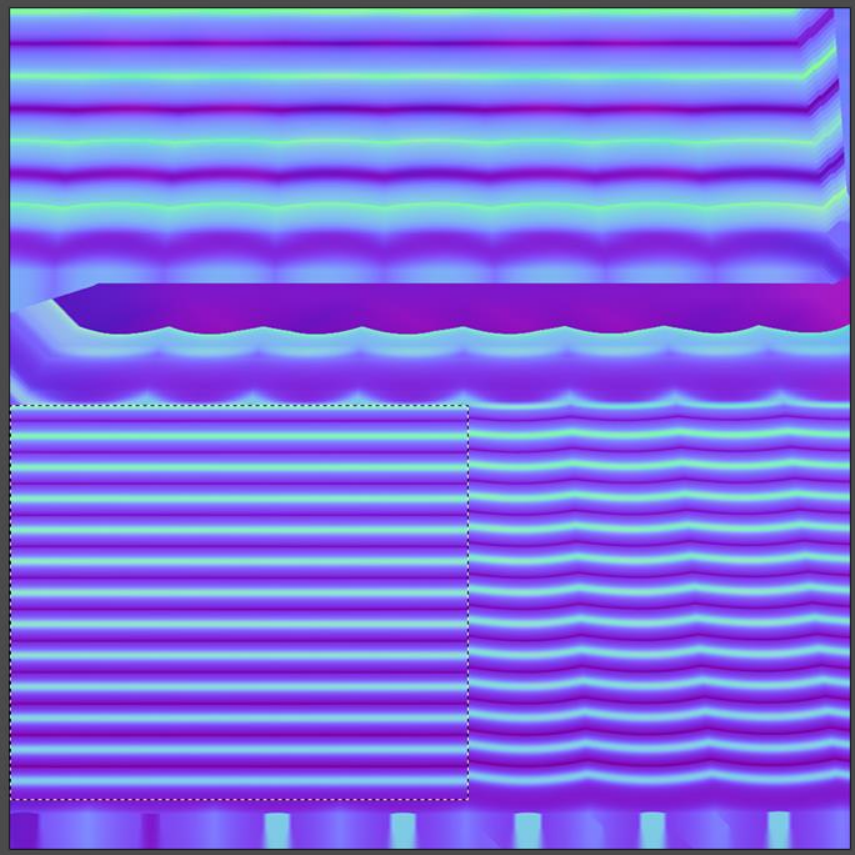
Editing Your Normal Map


NORMAL MAPS ARE NOT REGULAR TEXTURES
YOU CAN'T 'FIX' THEM IN PHOTOSHOP

EDITING YOUR NORMAL MAP

Painting out errors

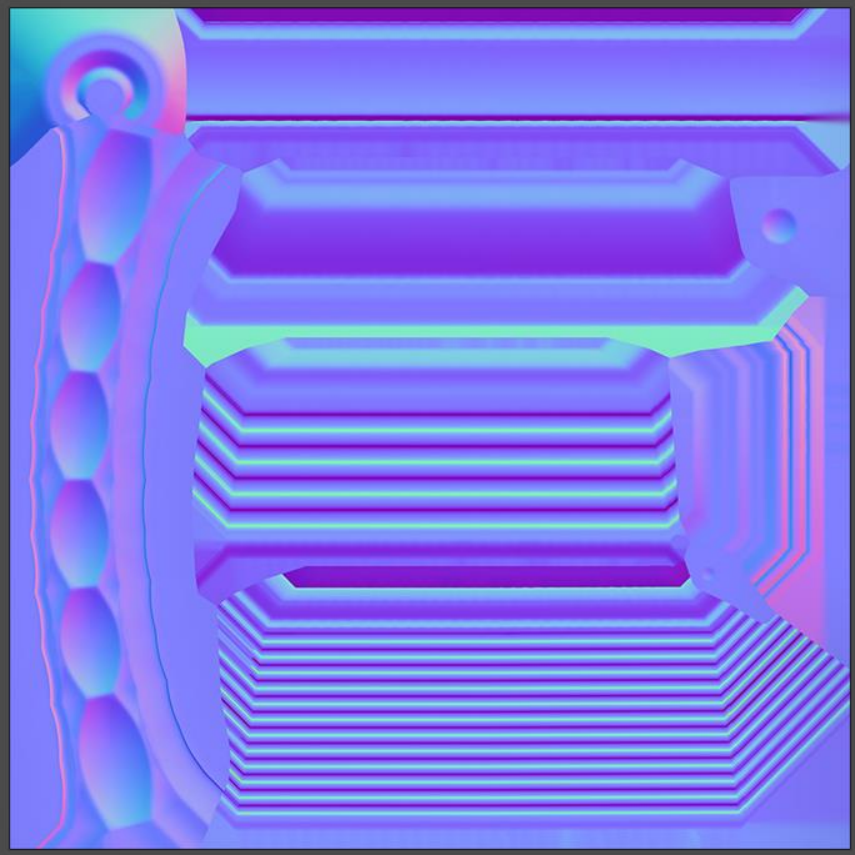
-Never paint out errors
-Normal Maps store mathematical values,
not just colour information
-Data is encoded a during bake via the Tangent Basis
-Editing this information directly modifies the vectors
-Most likely the vectors will be incorrect after editing
-Only certain colours represent correct normals
-Editing must be done again if you or a co-worker
must re-bake
-The same editing must be done on all baked maps
(Normal, AO, Heights etc.)
-Better to spend a little more time to get a good bake
that doesn't require editing.
-Learn why your bakes have errors
(too little/poor geometry, no cage etc.)
-Fix the cause & not just the symptom
-You will become a better artist for it! :)



EDITING YOUR NORMAL MAP

Resizing your Normal Map

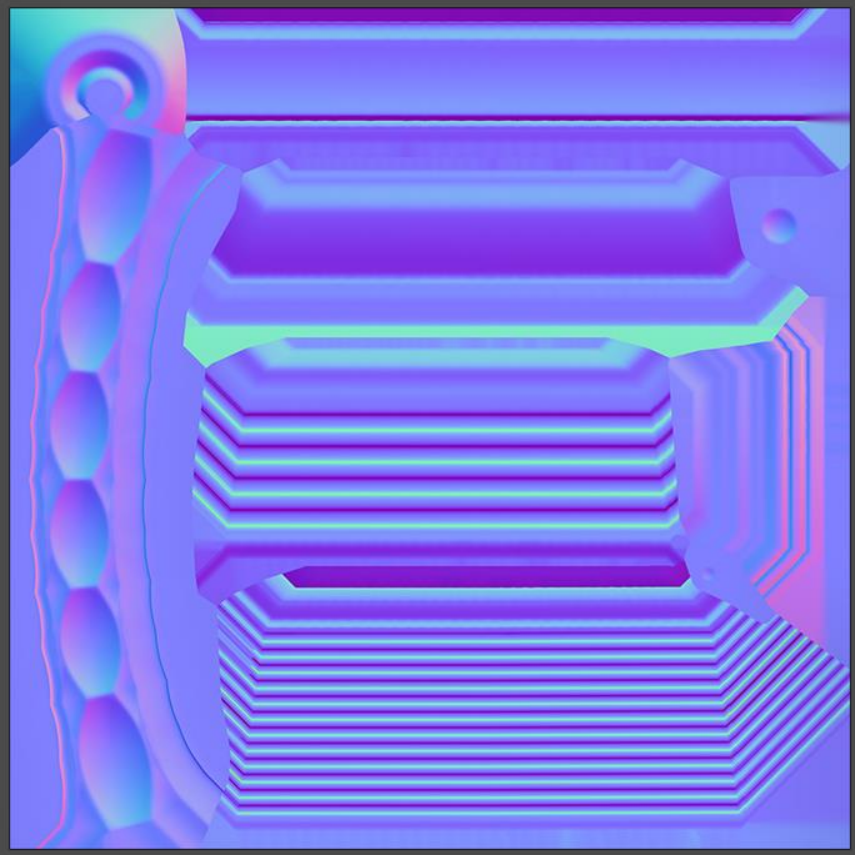
-Best to bake at native resolution if possible
-This is not always possible
-Different target platforms force smaller textures (PC, Playstation, Xbox, Wii, iOS, Android)



EDITING YOUR NORMAL MAP

Resizing your Normal Map

~~-Best to bake at native resolution if possible~~
-Bake as close as you can to the target resolution
-If you must resize use Bicubic interpolation
-Don't sharpen the resized result
-Always normalize after resizing
-Ensures that the normals are the correct length of 1
-Even though many shaders normalize automatically it is better to be safe
-If unsure check with coders



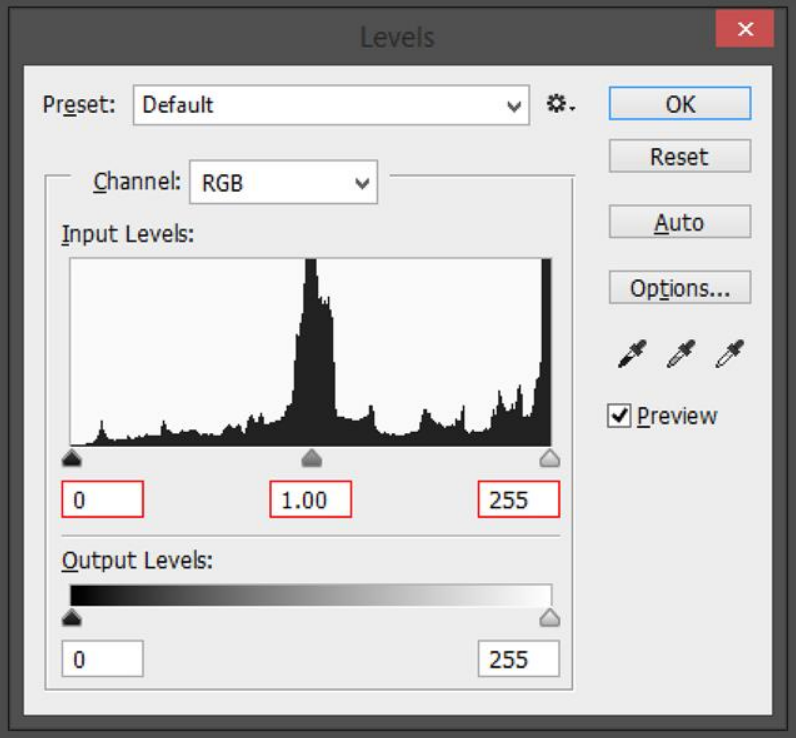
EDITING YOUR NORMAL MAP

Levels Adjustments

-Adjusting the levels in Photoshop can cause errors
-Data is encoded during the bake via the tangent basis can break
-Even a very small change can cause large errors
(mirroring, background colour etc.)

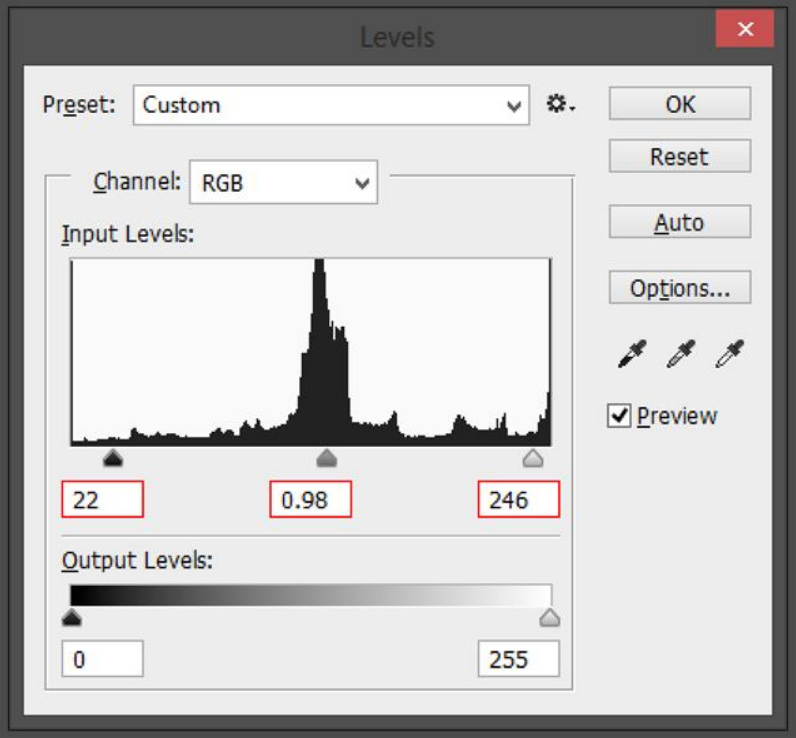
EDITING YOUR NORMAL MAP

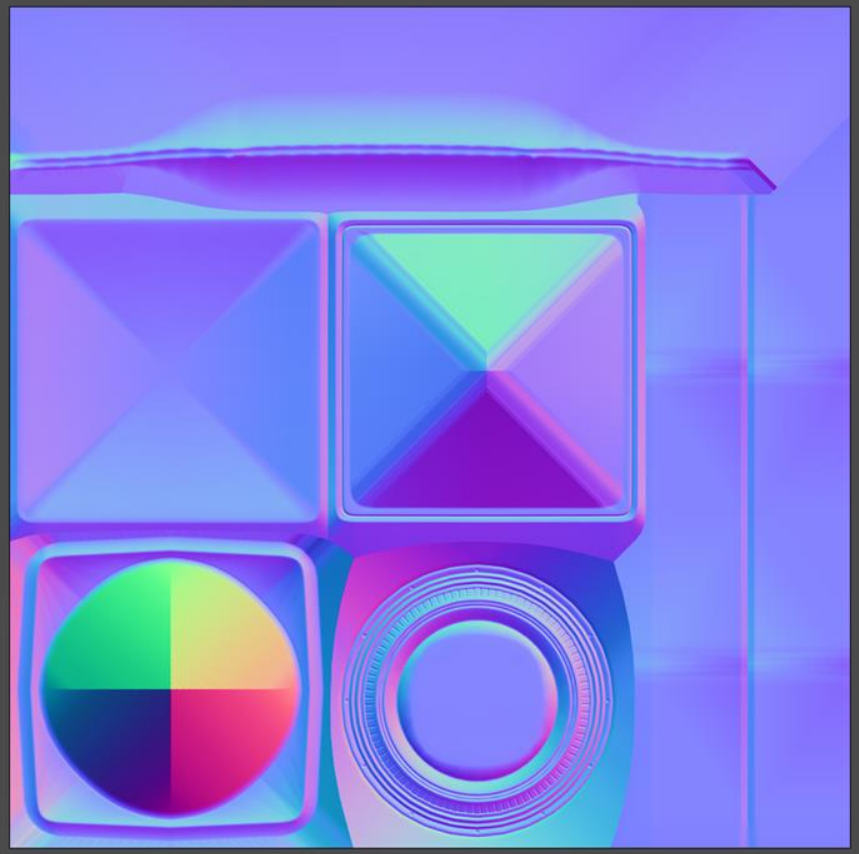
Unadjusted Normal Map



EDITING YOUR NORMAL MAP

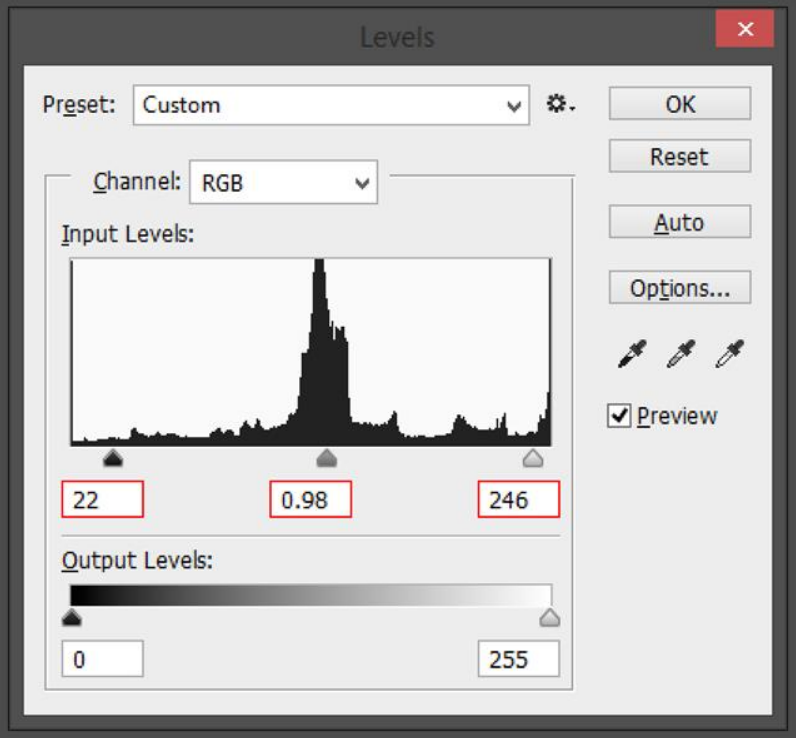
Levels Adjusted Normal Map





EDITING YOUR NORMAL MAP

Difference (Contrasted)



EDITING YOUR NORMAL MAP

Before

After

EDITING YOUR NORMAL MAP

Levels Adjustments

-If you need a stronger bake it's best to adjust the HP geometry
-Never edit your Normal maps in Photoshop!

EDITING YOUR NORMAL MAP

Levels Adjustments

-If you need a stronger bake it's best to adjust the HP geometry

~~-Never edit your Normal maps in Photoshop!~~

EDITING YOUR NORMAL MAP

Levels Adjustments

-If you need a stronger bake it's best to adjust the HP geometry
~~-Never edit your Normal maps in Photoshop!~~
-Normal maps generated from images don't always have the same limitations
-You still need to be careful and stick to some rules
-Ensure that the blue channel doesn't contain too much information
-Normalize the resulting map
-It's still better to change the strength of the initial normal map on generation

EDITING YOUR NORMAL MAP

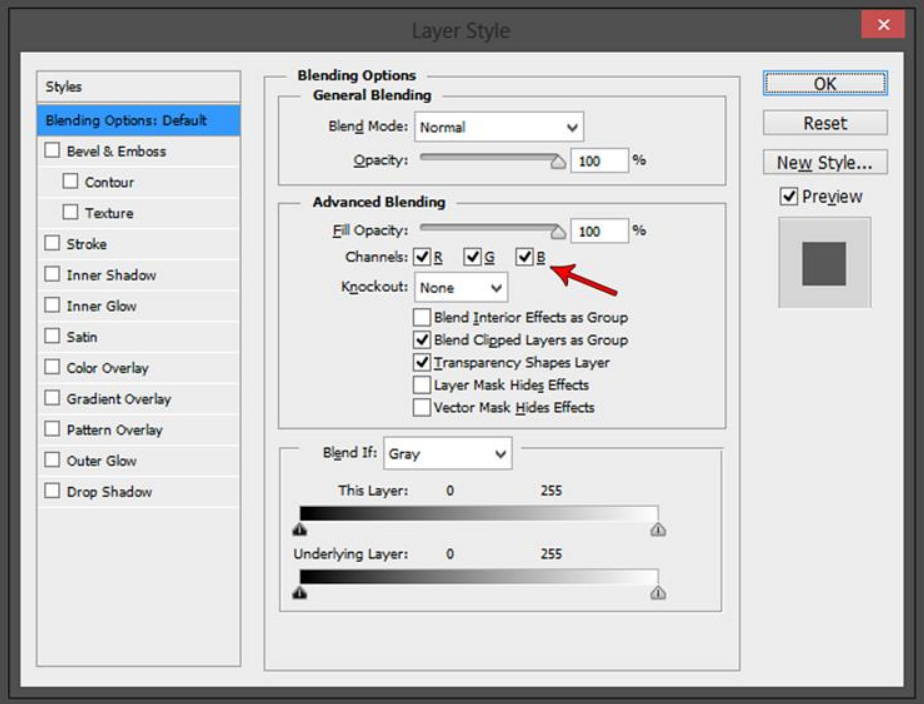
Adding details to your Normal Map after the bake

-This is acceptable :]
-Remove the blue channels information in the detail map

EDITING YOUR NORMAL MAP

Adding details to your Normal Map after the bake

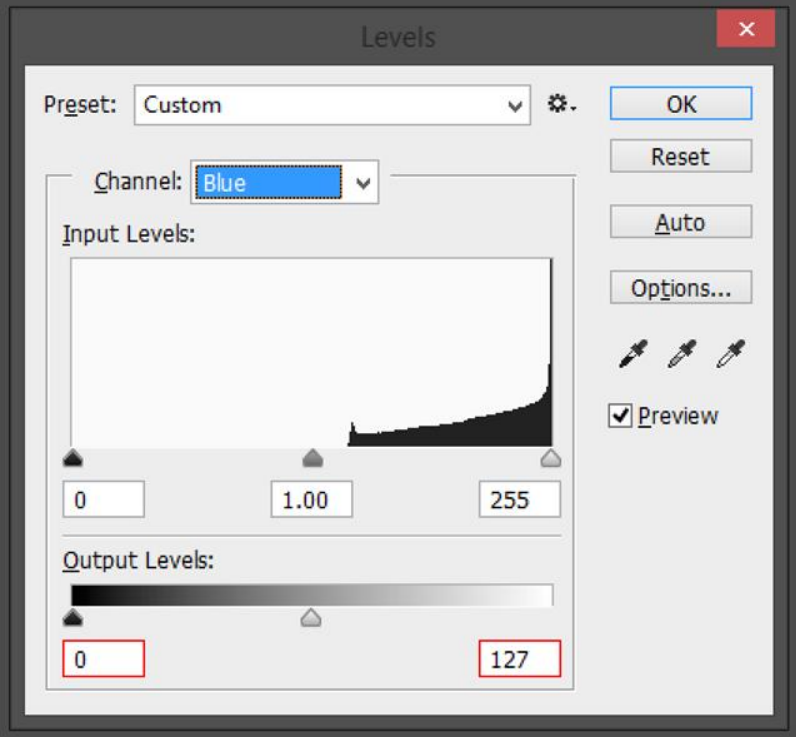
-This is acceptable :]
-Remove the blue channels information in the detail map
-Uncheck blue in the blending options



EDITING YOUR NORMAL MAP

Adding details to your Normal Map after the bake

-This is acceptable :]
-Remove the blue channels information in the detail map
-Uncheck blue in the blending options
-Add a levels Adjustment layer and set blue channels output to be 0-127



EDITING YOUR NORMAL MAP

Adding details to your Normal Map after the bake

-This is acceptable :]
-Remove the blue channels information in the detail map
-Uncheck blue in the blending options
-Add a levels Adjustment layer and set blue channels output to be 0-127
-Set to overlay blend mode
-Not accurate :[

EDITING YOUR NORMAL MAP

Adding details to your Normal Map after the bake

-This is acceptable :]
-Remove the blue channels information in the detail map
-Uncheck blue in the blending options
-Add a levels Adjustment layer and set blue channels output to be 0-127
-Set to overlay blend mode
-Not accurate :[
-This is the best way in Photoshop currently
-Normal map mixing is not suited to 2d

EDITING YOUR NORMAL MAP

Adding details to your Normal Map after the bake

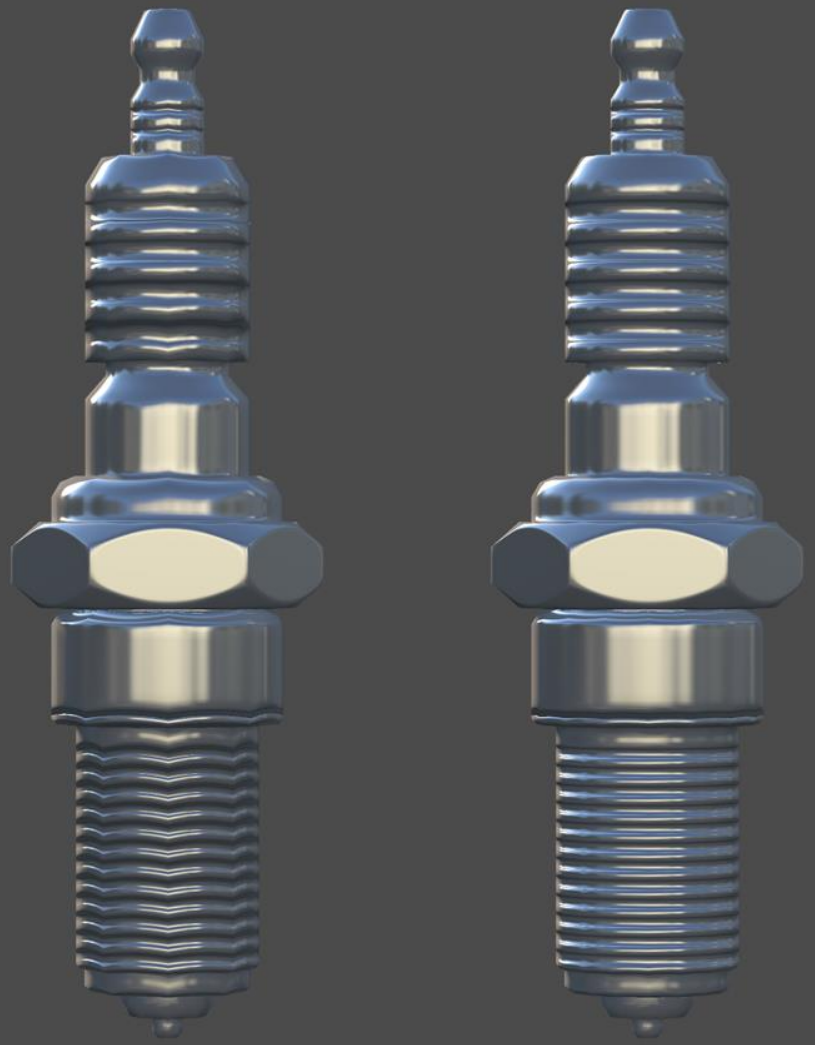
-This is acceptable :]
-Remove the blue channels information in the detail map
-Uncheck blue in the blending options
-Add a levels Adjustment layer and set blue channels output to be 0-127
-Set to overlay blend mode
-Not accurate :[
-This is the best way in Photoshop currently
-Normal map mixing is not suited to 2d
-Best done in an external application if possible.

Waviness & Skewing

WAVINESS & SKEWING

Waviness

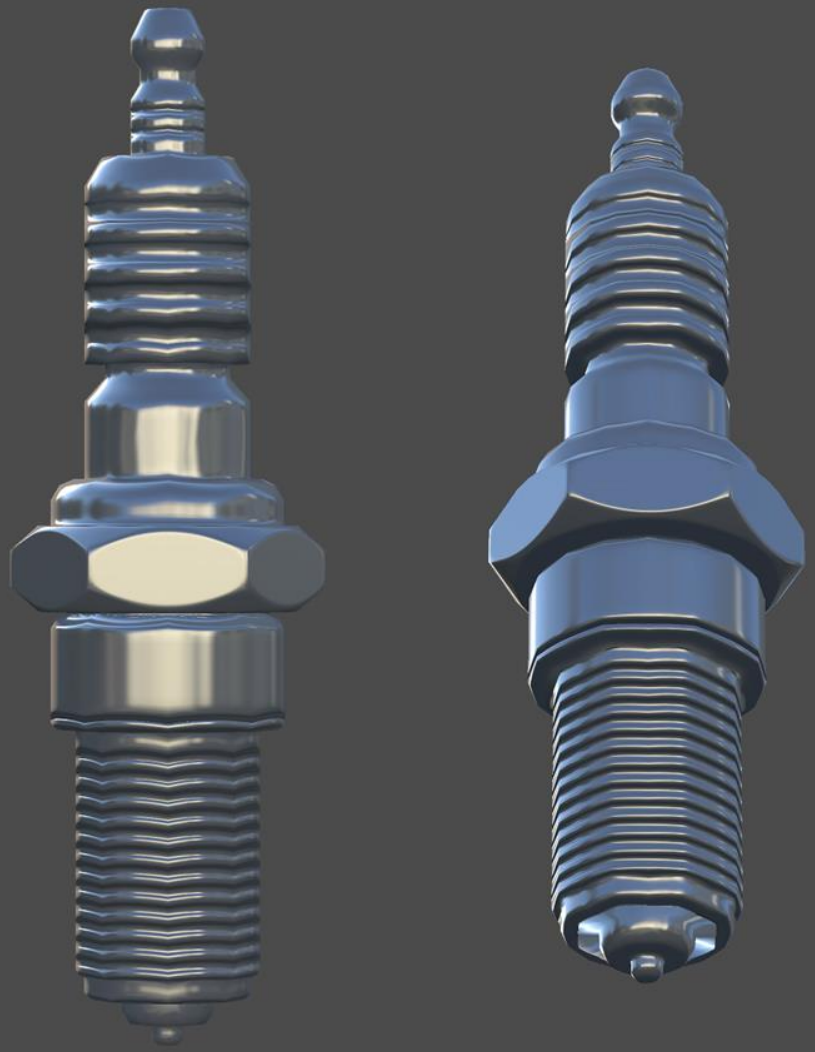
-Caused by a large difference between the high & low poly meshes
-Large difference equals more waviness



WAVINESS & SKEWING

Waviness

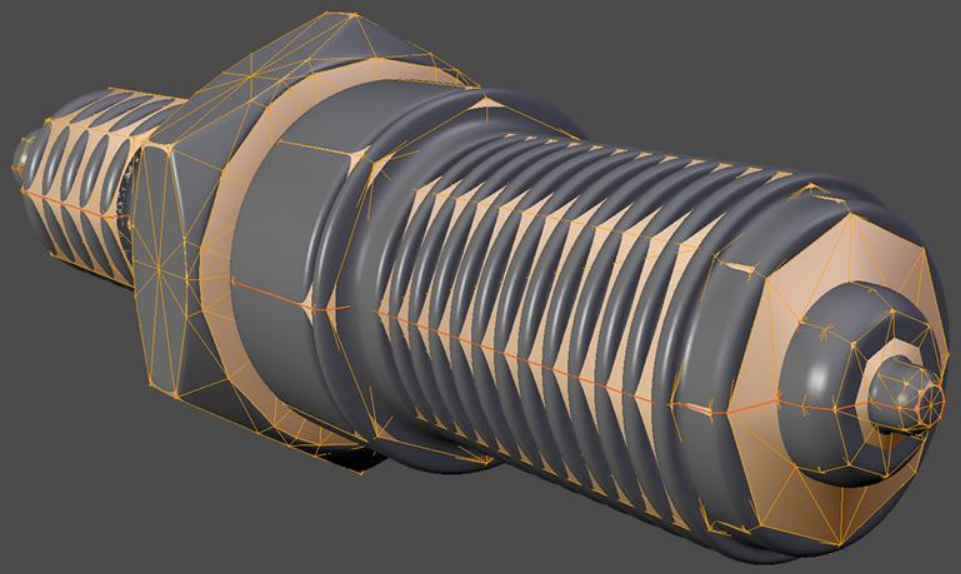
-Caused by a large difference between the high & low poly meshes
-Large difference equals more waviness
-Waviness isn't always bad
-The problem is view dependent
-Don't paint out waviness in Photoshop ;)



WAVINESS & SKEWING

Waviness

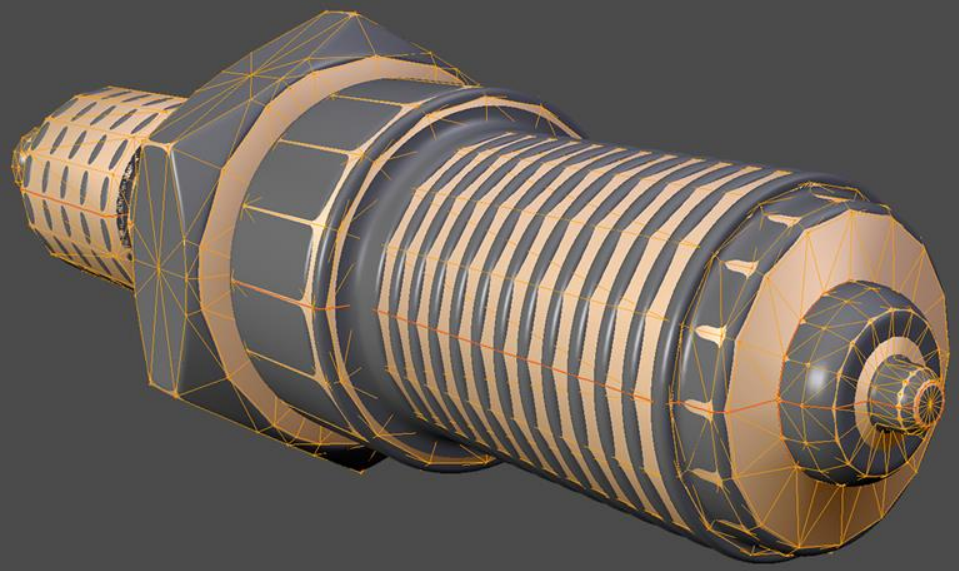
-Caused by a large difference between the high & low poly meshes
-Large difference equals more waviness
-Waviness isn't always bad
-The problem is view dependent
-Don't paint out waviness in Photoshop ;]
-Fixed by adding more sides to the low poly



WAVINESS & SKEWING

Waviness

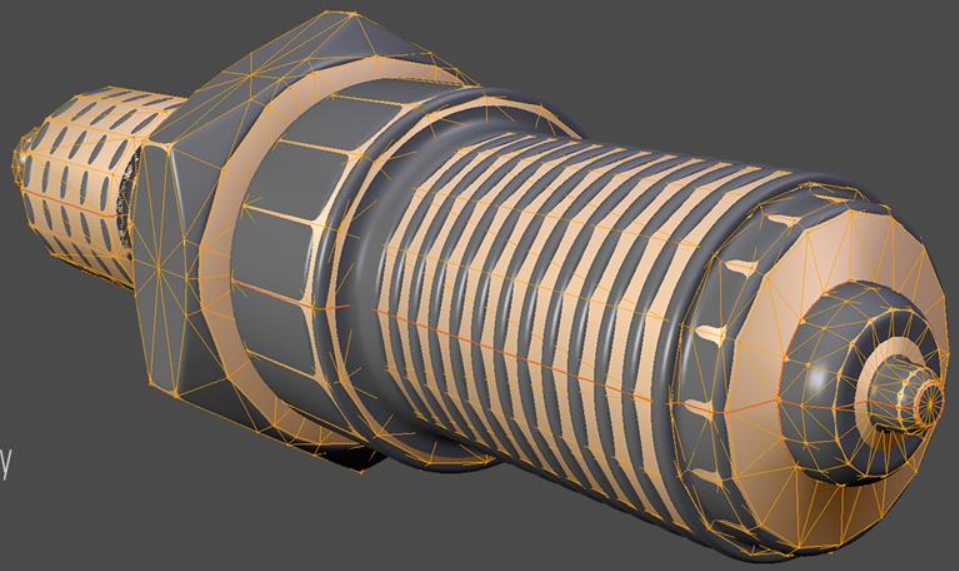
-Caused by a large difference between the high & low poly meshes
-Large difference equals more waviness
-Waviness isn't always bad
-The problem is view dependent
-Don't paint out waviness in Photoshop ;]
-Fixed by adding more sides to the low poly



WAVINESS & SKEWING

Waviness

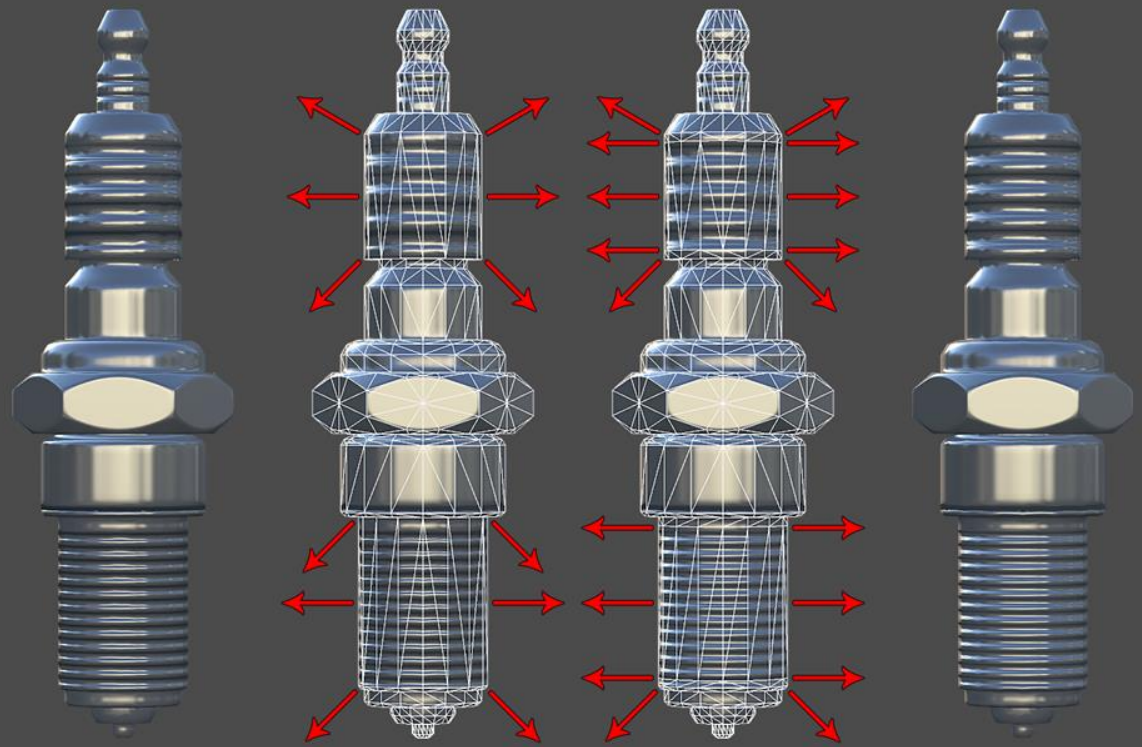
-Caused by a large difference between the high & low poly meshes
-Large difference equals more waviness
-Waviness isn't always bad
-The problem is view dependent
-Don't paint out waviness in Photoshop ;]
-Fixed by adding more sides to the low poly
-Budget doesn't allow for more geometry?
-Bake to object space
-Remove the extra sides whilst maintaining the UVs
-Convert to tangent space with new lighter geometry



WAVINESS & SKEWING

Skewing

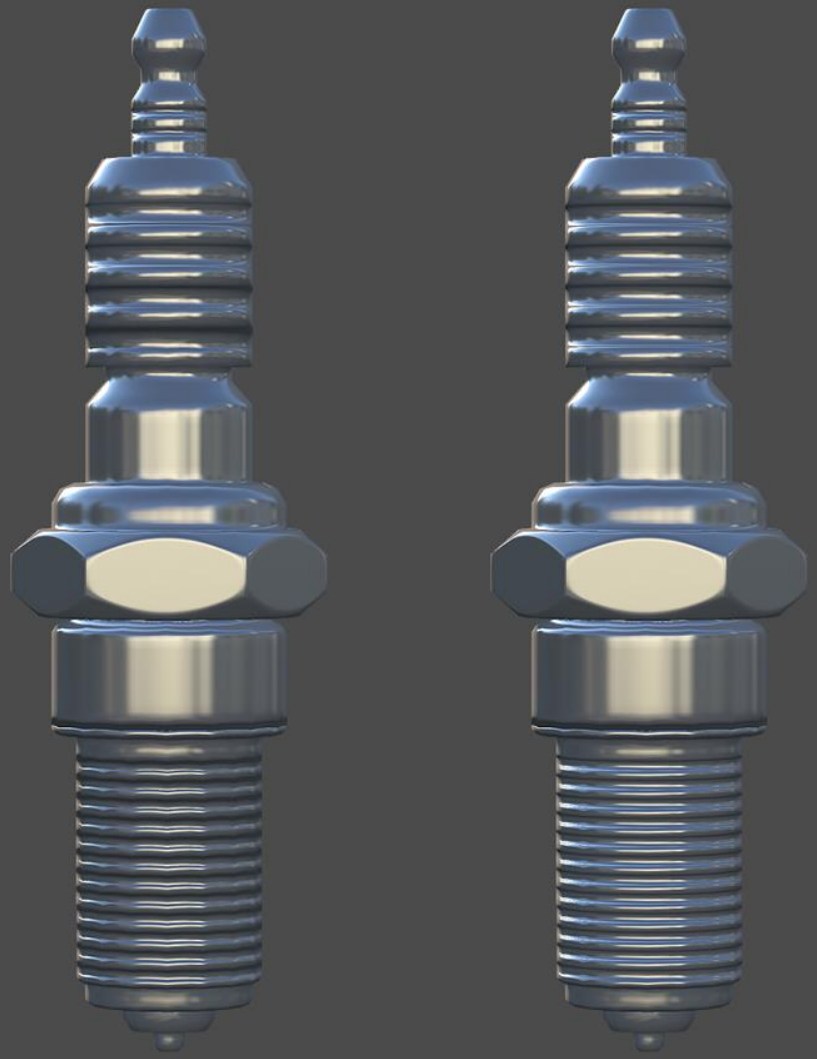
-Normals are averaged along the length of the mesh
-The projection is variable which results in skewing
-Adding control loops flattens the normals
-The resulting bake is even



WAVINESS & SKEWING

Skewing

-More even projection
-Less waviness in the bake
-More consistent with the original high poly geometry
-Specular highlights are more even
-Less waviness in the bake
-Less aliasing in final result



SPECIAL THANKS

Thanks to

Joe Harford - www.airship-images.com

Questions?