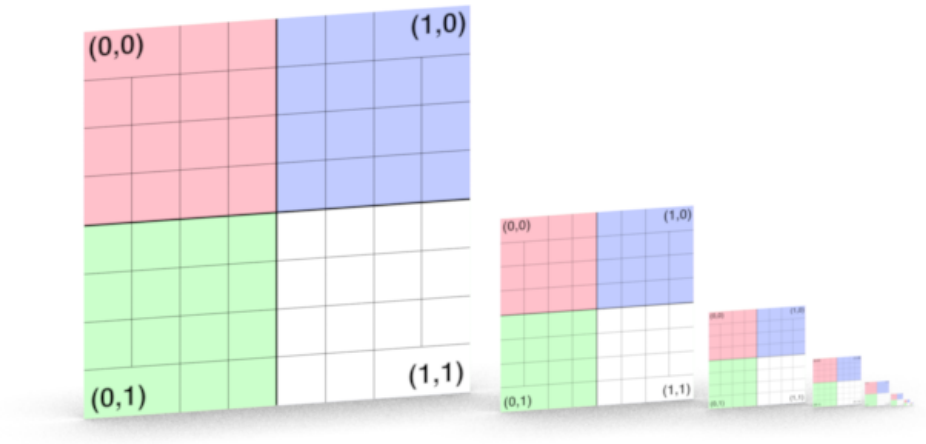
# txmake





- Texture
- Mipmapping
- Environment Map
- Format
- Bump To Roughness
- Shadow Map (deprecated)
- Command Line


txmake reads an image file (or multiple image files) called *foo.format* and creates a RenderMan texture file called *foo.tex*. The previous contents of *foo.tex* (i f any) are silently overwritten. The following formats are supported for input files: BMP, DPX, GIF, HDR, JPEG, MayaIFF, OpenEXR, PIC, PNG, PPM, Pixar DeepData, Pixar Texture, SGI, TGA, TIFF, XPM, Z-File

txmake was written based on the IceMan image compositing engine, adding support for the following features: the ability to operate on large input images, all image computations are performed in floating point, multiple channels can be selected, and filters can now be applied as each successive level in a MIPmap or RIPmap are computed.

## Texture

### Wrap Mode

-mode <m>  Applies the <m> setting choice to S and T
-smode <m>  Applies the <m> setting choice to S
-tmode <m>  Applies the <m> setting choice to T

<m> Controls the texture values returned for texture coordinates outside the first ST tile.

- **black** Useful for tile based workflows such as UDIM.
- **clamp** Useful non-periodic texturing access.
- **periodic** Useful for periodic texture access.

Mode black
Mode clamp
Mode periodic

The default is 'black'.

### Image Resize

-resize <r>  Applies a resize operation to the nearest power-of-two size.

For simple textures or environment textures the resolution of the texture file is determined from the height and width of the image file and the operation specified by the resize option. If either the width or the height of the image is an exact power of two, that dimension is left unchanged. Any dimension that is not an exact power of two will be adjusted according to the resize operation. If the image is already a power of two in dimension, its size will be unchanged. There is no size limit for conversion other than the file type used and the machine resources performing the conversion.

Note that many file formats may have limitations on their storage type and size independent of txmake. TIFF and EXR are the most common types.

- TIFF files converted to .tx may have better performance than other formats but are limited to the maximum size of 4GB
- OpenEXR files may be larger and potentially slower but these have the fewest restrictions when converted to .tx as *-format openexr*

<r> The resize operation may be one of the following:

- **none** Do not resize the image. The texture map size will be the next higher power of two. Texture file area not covered by the image will be set to black.
- **up** Resize the image up to the next higher power of two. Texture access will correct for the resize.
- **down** Resize the image down to the next lower power of two. Texture access will correct for the resize.
- **round** Resize the image to the nearest power of two. Texture access will correct for the resize.
- **up-** Resize the image up to the next higher power of two. Texture coordinates will be 0 to 1 across the resulting image.
- **down-** Resize the image down to the next lower power of two. Texture coordinates will be 0 to 1 across the resulting image.
- **round-** Resize the image to the nearest power of two. Texture coordinates will be 0 to 1 across the resulting image.

Resize none
Resize up
Resize up-

*Image credit NASA*

The default is 'up'.

### Resize Filter

-filter <f>  Applies one of the follow filter choices when resized.

<f> Filter used to resize the image. The below options are available for resizing an image. Sharpening filters tend to be preferred for enlargements.

- point
- box
- triangle
- sinc
- gaussian
- gaussian-soft
- catmull-rom
- mitchell
- cubic
- lanczos
- bessel
- blackman-harris

The default is 'catmull-rom'

### Image Blur (-blur <f> / -sblur <f> / -tblur <f>)

-blur <f>  Applies the blur to both S and T directions
-sblur <f>  Applies the blur to S only
-tblue <f>  Applies the blur to T only

<f> Values greater than 1.0 blur the image.

The default is 1.0.

### Bleed Alpha

-bleedalpha

Radially dilate pixel values around alpha mask, useful for texture baking. This gutter helps avoid sampling artifacts around ST seams.

## Mipmapping

Mipmapping is used to improve texture aliasing and rendering performance by automatically choosing an appropriate texture resolution level from a pyramid based on viewing size and angle of the texture on screen. This can reduce processing time at the cost of details that would go unnoticed.

### Mipmap Pattern

-pattern <p>

<p> can be one of the following:

- **single** Only a single texture image is stored in the texture file. This effectively disables mipmapping and will result in very poor texturing performance for most applications.
- **diagonal** Lower resolution textures are generated by successively reducing the texture image to one-half of its resolution in both the S and T directions. This results in a texture file which is approximately one-third larger than the original texture image.
- **all** A complete set of texture images is stored in the texture file (i.e. ripmap). This will make the texture file approximately four times as large as the original texture image.

The default is 'diagonal'

### Mipmap Source

-mipfrom <s/n>

<s/n> can be one of the follow where using *n* is an integer.

- **previous** Create each mipmap level from the preceding level.
- **source** Create all mipmaps from the source image.
- **<n>** Specify explicit mipmap level

The default is 'source'

### User Mipmaps

-usermipmap <picnames>

The files in <picnames> correspond to the levels in the pyramid (first name in picnames = highest resolution level of the pyramid). Note that there are several rules for generating these "user mipmaps":

- The first image determines the transformation matrix to be put into the texture.
- The first image determines the texture's dimensions.
- All images listed in picnames must have the same number of channels.
- An image listed in picnames must be at least as large as the corresponding pyramid level (i.e. a 4x4 image can't go into an 8x8 image).
- Images in picnames must all be the same relative dimensions (i.e., a 4x2 image and a 4x4 image are not compatible, but 4x4 and 8x8 are compatible as are a 4x2 and a 8x4.)
- Unless the format is tiff, no image listed in picnames can be larger than the first image.
- Unless "singl" is chosen, then the last image listed in picnames will be filtered up the rest of the pyramid.

### Mipmap Filter

-mipfilter <f>

<f> is the pixel filter used for mipmap creation. Select from the below:

- point
- box
- triangle
- sinc
- gaussian
- gaussian-soft
- catmull-rom
- mitchell
- cubic
- lanczos
- bessel
- blackman-harris

The default is 'box'.

## Environment Map

### Lat-Long Map

-envlatl

Make a lat-long environment map.

### Diffuse Convolution

-diffconv <brdf,res,mip>

Txmake can produce side-band data for use in the control variates method of Monte Carlo integration. The side-bands are the convolution of a diffuse BRDF with an environment map. Only lat-long maps are supported. OpenEXR and TIFF files are supported.

- *brdf* is one of {Lambert, KajiyaKay}. This is required.
- *xres* is the image width in a mip-map pyramid at which the convolution will be computed. This is optional and defaults to 256.
- "mip" is optional and literal. If added, the convolution image will be mip-mapped, otherwise not (the other mip levels will contain black).

### Cube Map

-envcube

Make a cube environment map from 6 images, e.g.

```
txmake -format openexr red.png yellow.png green.png cyan.png blue.png magenta.png cubemap.exr
```

Environment cube maps are created with format specific layout:

format tiff
format pixar
format openexr

### Filed of View

-fov <f>

<f> is a cube map field-of-view in degrees.

### Ptex Cube Map

-penvcube

Make a ptex cube environment map from 6 images

### Spherical Harmonic Bands

-shbands <n>

Embed <n> spherical harmonic bands.

## Format

### Image Format

-format <f>

<f> corresponds to the underlying texture image format (openexr is required for large, >4GB, resulting images):

- **tiff**
- **pixar**
- **openexr**

A default of tiff is specified as a configuration option in the renderman.ini.

### Data Compression

-compression <c>

For the Pixar format <c> is:

- none
- lossless
- lossy

For OpenEXR format <c> is:

- none
- rle
- zip
- piz
- pxr24
- b44

- b44a

The TIFF format always uses LZW compression. Pixar format defaults to 'lossless' compression while OpenEXR defaults to 'zip'.

### Data Type

-byte / -short / -half / -float

Control the file format datatype.

- **byte** outputs unsigned 8-bit data
- **short** outputs signed 16-bit data
- **half** outputs half-float data (OpenEXR/penv)
- **float** outputs float data

### Channel

-ch <#,#:#>

By default, the number and order of channels in the texture file depends directly on the set of channels in the image file. An RGBA picture will generate a four-channel texture, an RGB picture will generate a three-channel texture, and an R picture will generate a one-channel texture. The -ch option can be used to select and reorder the channels of the image file. The channel #'s are used to select the channels of the image file. Channels indexing begins at 0. For RGB, `-ch 1` will select only the green channel of the image file and puts it into channel 0 of the texture file. A list of numbers can be used to select multiple channels, for example, `-ch 0,1,2` and `-ch 0:2` will select the RGB channels of an image file while `-ch 2,1,0` will select those same channels but in order blue, red, green.

### Display Window

-extraargs displaywindow

displaywindow only for exr input

## Bump To Roughness

Bump to roughness requires all bump or normal maps to be converted to a *.b2r texture.

b2r textures contain 5 channels and are always generated in OpenEXR format as it provides the best compression ratios for half-float data.

-bumprough <f> <i> <n> <iu> <iv> <i>

- <f> : roughness scaling factor
  - Default: 2.0 (very rarely needs to be changed)
- <i> : invert bump signal (0|1)
  - Default: 0
- <n> : interpret as normal map (0|1)
  - Default: 0
- <iu> : invert map in U (0|1)
  - Default: 0
- <iv> : invert map in V (0|1)
  - Default: 0
- <iv> : refit slope distribution for better results.
  - Default: 1.
  - This step is time consuming but preserves details much better when seen from afar.

A typical txmake invocation for a bump map:

```
txmake -resize round- -mode periodic -filter box -mipfilter box -bumprough 2 0 0 0 0 1 bump.tif bump.b2r
```

For a normal map:

```
txmake -resize round- -mode periodic -filter box -mipfilter box -bumprough 2 0 1 0 0 1 normalmap.tif normalmap.
b2r
```

⚠ The filter and MIP filter need to be set to "box" to ensure the math works correctly.

## Shadow Map (deprecated)

### Shadow

-shadow

Make a shadow map from a z-image.

### Min/Max Shadow Map

-minmaxshadow

Make a minmax shadow map.

### Pyramid Shadow Maps

-pyramidshadow

Make a pyramid shadow map.

## Command Line

### Newer

-newer

Do nothing if source is older than target texture.

### Directory Map

-dirmap <s> -dirmapend

Apply dirmapping to input and output files. Multiple dirmaps can be specified. Syntax for <s> should look similar to RiOption "searchpath" "dirmap", e.g.:

[\"NFS\" \"z:/\" \"/mnt/\"] [\"NFS\" \"Z:/\" \"/mnt/\"]

### Threads

-t:X

Use X threads. The default uses all threads.

### Verbose

-verbose

Print updates as txmake executes.

### Version

-version

Print version and exit.

### Help

-help

Print usage and exit.