



Sparrowhawk 3D

The Auto-MipMapper Shader

for CARRARA 4,5 & 6

Version 0.2

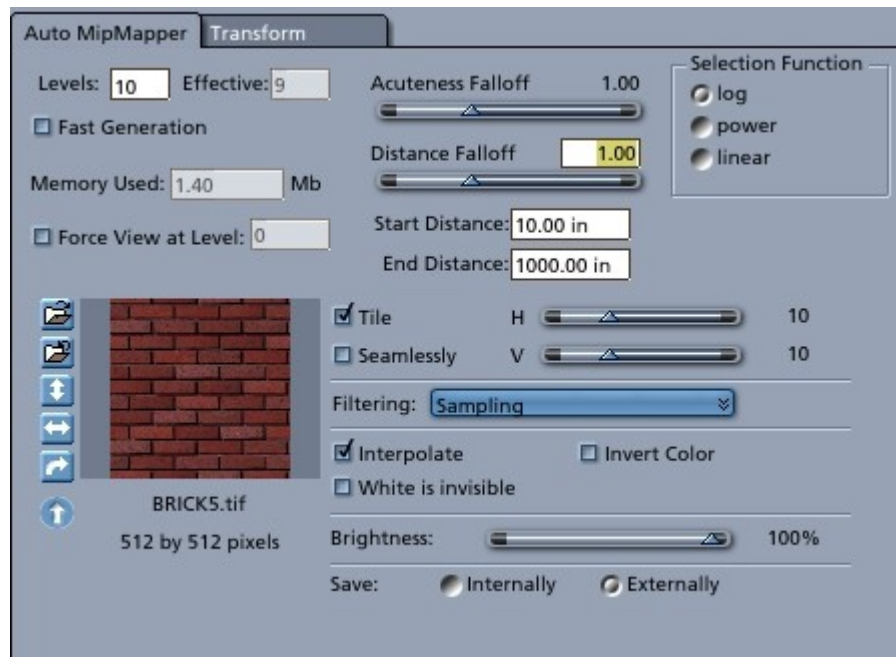
The Auto-MipMapper reduces or removes moire patterns and twinklies in both still renders and animations. Unlike the MipMapper the Auto-MipMapper generates the required mipmaps and stores them in memory when the scene or a new texture map is loaded. Mipmaps are not saved with the scene file and only have a 33% additional memory cost in addition to the source texture map. Scenes using many mipmapped shaders may be slow to load or appear to hang while the maps are generated.

To install the plugin extract the contents of the compressed zip package into the Carrara\Extensions sub-folder. The plugin must also be installed into the ..\Extensions folder of all render nodes. To remove the Auto-MipMapper Carrara 4 users must delete the AutoMipMapper.mcx and AutoMipMapper.dta files from the ..\Extensions\ folder. Carrara 5 and 6 users must delete the AutoMipMapper.mcx,AutoMipMapper.dat and AutoMipMapper.txt files.

Macintosh users should open the Carrara installation folder (typically /Applications/Carrara). Right-click on the Carrara or Carrara Pro application and then choose "Show Package Content". Navigate to "Contents/MacOS/Extensions" then copy the AutoMipMapper.mcx, AutoMipMapper.dat and AutoMipMapper.txt files into that folder.

The Auto-MipMapper User Interface

Once installed the Auto-MipMapper shader can be found in the main shader list or under the Sparrowhawk3D submenu. It can be applied to any sub-shader or channel.



Levels: Set the number of mipmaps to generate. The first mipmap is half the resolution of the source texture map and then each successive mipmap is half the size of the previous level.

Effective: Gives the recommended number of effective mipmap levels. For example, from a 512x512 pixel source texture map the 1st level will be 256x256, the 2nd 128x128, and so on until the 9th map which is 1x1 pixel. Any maps generated after that are always 1x1 pixel and ineffective in the selection algorithm. Begin with a large number of levels then set the parameter to the recommended effective levels.

Fast Generation: When enabled this speeds up the generation of the mipmaps by using the previous map to generate the next map. For some source textures this can introduce artefacts, particularly with repeating tile patterns such as bricks at the seams. Normally every mipmap is generated by smoothly scaling down the original source texture map.

Force View at Level: When checked this disables the mipmapper. If the force level is set to 0 the source texture map is applied. Use this to check what any of the generated maps look like by setting the preview window to view a flat preview.

Acuteness Falloff: When the textured surface of an object is viewed side on, for example looking down a long brick wall or corridor, then tiled textures will render with moire patterns (waves). Increase the Acuteness Falloff to bring the lower resolution maps into play earlier. Reduce the Acuteness Falloff if the maps appear to be too chunky or blurry when viewed side on.

Distance Falloff: As the textured surface gets further away from the camera the lower resolution mipmaps are selected and blended in. Increase the Distance Falloff if moire patterns or twinkies are present, reduce it if the texture is too chunky or blurry.

Selection Function: The default logarithmic selection of mipmaps based on the log of the distance from the camera gives the best results. The other functions are provided for user experimentation.

Start Distance: When a surface is viewed front-on and closer than the Start Distance then only the original source texture map is used. Set the Start Distance so that when the textured surface is viewed up close at the final render resolution all the detail is clear. Increase the Start Distance for low resolution textures on small objects.

End Distance: The End Distance is not used when the Selection Function is set to 'log' mode. Increase the End Distance when using the power and linear Selection Functions and any surface rendered at or beyond the End Distance will always use the last mipmap level.

Memory Used: Textures loaded with the mipmapper do not require much more memory than the source texture since each successive map is half the size. The memory readout is there to help manage texture memory since textures loaded with the Auto-MipMapper are not added to the render statistics. The memory used by each instance is given, not the total memory used

by all shaders with mipmaps.

The standard texture map shader UI appears at the bottom of the shader. Loading or reloading the texture map triggers a re-generation of the mipmaps. Set the filtering to sampling for faster renders. Check the Interpolate option for the best results with the Auto-MipMapper as this will blend the values of the lower resolution maps and give much smoother renders for animations.

Using the Auto-MipMapper

Some patience and testing is required on the part of the user to adjust the Auto-MipMapper settings for optimal results. Animations can be improved by bringing the mipmaps in nearer and having slightly blurry textures rather than annoying patterns and twinkly artefacts - its a trade off. Do not expect to get perfect results but the Auto-MipMapper can make great improvements to the quality of your renders. Best results always come from setting the renderer to use the highest level of anti-aliasing.

The Auto-MipMapper parameters need to be set so that when the camera is close to the textured surface no detail is lost and when the object is seen at a distance the last 1x1 pixel mipmap is applied. Adjust the shader preview window size then track the camera up close to the object. If the surface is blurry or chunky then increase the Start Distance. Now pull the camera back until the textured surface of the object is only a few pixels small in the preview then use the 2D Zoom tool to click on it a few times and look at which mipmap is being used. In the default log mode adjust the Distance Falloff until the last mipmap levels, ie the 2x2 and 1x1 pixel texture maps, are used - the surface will appear with no texture and a single colour. In the power or linear selection function modes adjust the End Distance until the 1x1 pixel texture map is selected. From the top menu choose View | Fit Object in Preview then track the camera back and forth to various distances. When applying the Auto-MipMapper to tiled surfaces like bricks increase the Distance Falloff setting if moire patterns are still appearing.

The complex factors of the final render resolution, the texture resolution and how it is wrapped or tiled over a surface all come into play. The size of the rendered texture not the object is what really matters. A small object with a very high resolution texture is different from a large one. A low resolution animation intended for the web is different from a DVD quality render or a high resolution still image. Adjust the shader preview window size so that it matches the size of your final render when using lower anti-aliasing settings and set the preview window size to double the final render size for animations that use the best anti-aliasing. Turn off global illumination and turn down the render quality when rendering still images and adjusting the Auto-MipMapper settings.

Moire patterns in tiles and twinkling artefacts in animations happen when there are more pixels in the source texture than appear in the rendered image. For example if a 256x256 texture map is applied to a brick wall and then rendered up very close, so that the wall fills the view, then there is not quite enough detail in the source texture for a default render size of 640x480. If the texture map parameters are set to interpolate Carrara will smoothly fill in the gaps, if not the texture will appear to have chunky pixels. If the camera pulls back so that the same wall fills about half the view then the texture will render close to its original 256x256 size. Pull the camera back further and the same texture will now be rendered at a fraction of its original size, say 42x42 which means that Carrara's ray tracing engine will pick out only some of the original source pixels - move the camera and it will pick out a different set of pixels and so they twinkle.

Moire patterns exist in reality, walk past a picket fence on a corner and you'll see them. A newsreader with a poor choice of patterned tie has them. Even in the best DVD and Blu-ray movies landscapes will twinkle away in the background.

MipMapped objects seen in reflective or through refractive surfaces will render with inconsistent results. The Auto-MipMapper uses a direct measure of the angle and distance from the camera to the shaded surface.

Examples

This example uses a very simple and low detail miniature house that was part of a set used to create a whole suburb. Without the Auto-MipMapper the roof and weatherboards produce strong moire patterns. These ripple like patterns are ugly in still renders and they flicker and spoil the animation. A real fixer-upper. The image from the preview window on the right uses the Auto-MipMapper, it may appear to be a little blurry and loose detail but where would you rather live ?



The next example uses a more detailed building with a repeating brick texture. In both images the anti-aliasing has been set to fast to highlight the problem. At the same distance the moire patterns down the right hand side of the building arent noticeable when the best anti-aliasing is used for a still image but when the building is animated...



Tips and Tricks

- The bump channel is not affected by the Auto-MipMapper. Fractal bump effects used on terrains and other geometry will still twinkle. Bake a fractal colour or bump map and use it with the Auto-MipMapper for distant terrain objects. *I have spin-off plugins that can be used with a multiply shader node to drop the value of a bump map.*
- If the Auto-MipMapper is applied to a figure or object that has an unfolded or seamed style texture layout, eg a Daz/Poser figure or costume, then set a low number of mipmap levels to avoid the seams becoming visible. Unless there is a generous overlap around the edges of the layout then as the source texture shrinks the white spaces between the layout will show up as seams.
- Render a few frames of an animation to test if there are too many twinklies.
- Use a noise filter in your video editing software to get rid of even more twinklies.

Planned Improvements

- More user friendly and faster method to set the parameters - if one can be found
- Apply filtering functions in the mipmap generation code
- A menu command to convert existing texture maps of a shader into mipmaps
- Add an button to work out the number of effective mipmaps and automatically set it

Support

Sparrowhawke3D is an amateur project so I can't make any promises regarding support, however I'm keen to improve my simple practical plug-ins and share them with the Carrara community.

Special thanks to Untoldwind for the Carrara 6 builds for the Macintosh.

The Auto-MipMapper plugin is absolutely free, you can use it in any project, commercial or otherwise. You may not repackage and sell the Auto-MipMapper.

For comments, feedback and suggestions email me, Jeremy@Sparrowhawke3D.com.