RE:Map Documentation

RE:Map is a set of 5 plugins: RE:Map Distort, RE:Map Displace, RE:Map UV, RE:Map Inverse UV and RE:Map CornerPin.

Please note we often update our products, please consult our website for the latest. You will also find a growing set of Tutorials on our website to complement the documentation. Also note the snapshots of the user interface might slightly differ between applications as certain applications support different functionalities and others not.

What are the RE:Map plugins?

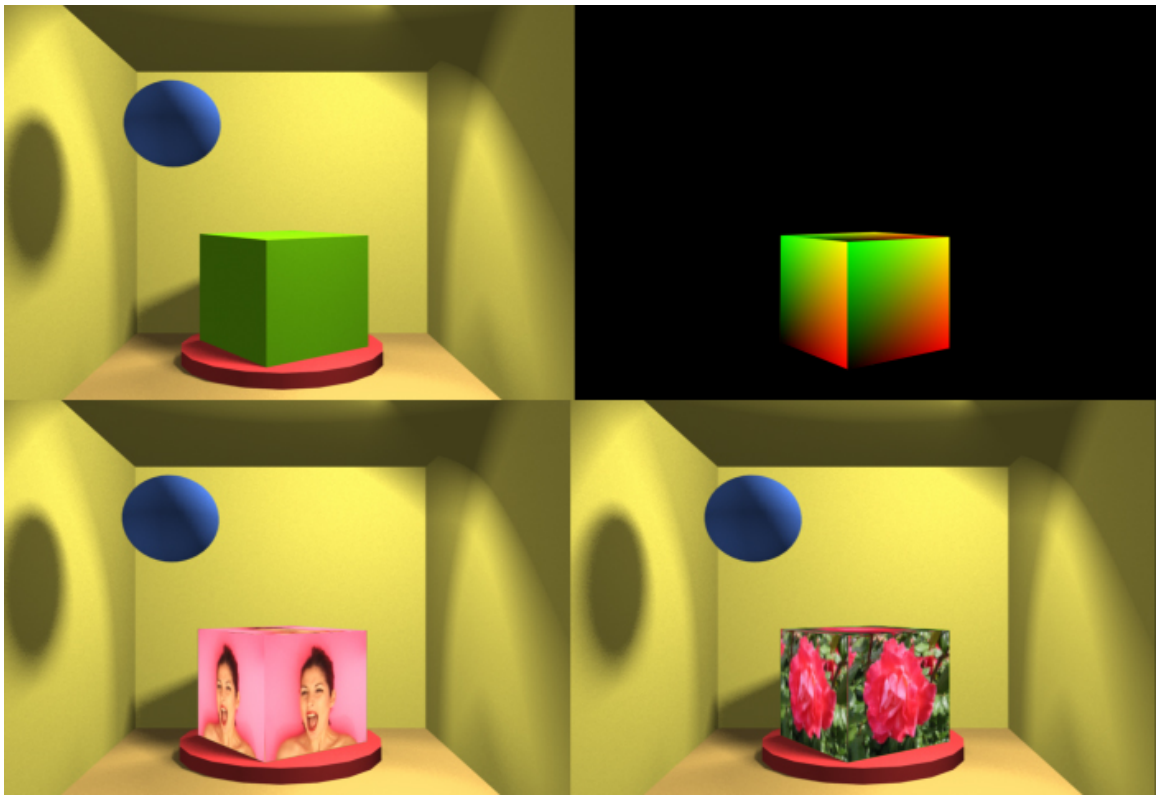
* RE:Map **Distort** takes a color image and automatically distorts the image based on the features for a caricature-like effect.
* RE:Map **Displace** warps an image with a user-supplied displacement map. This plugin includes many more options than most other displacement map filters; notably, RE:Map Displace has advanced filtering controls for better results.
* RE:Map **UV** takes a UV Map, probably rendered from a 3D system, and renders an image using this mapping. Easily clean up texture maps and remap in post without having to go back to your 3D system for re-rendering. In addition, this plugin allows you you to come up with cool 2D and 3D animated texture map templates and reuse them over again with different image sequences.

* RE:Map **Inverse UV** takes a UV Map and a color image that matches that UV Map and inversely projects the color image to a rectangular image that may then be used to texture the UV Map with RE:Map UV, or for use as a texture map for your 3D system.
* RE:Map **Corner Pin** allows one to project (corner pin) or unproject (inverse corner pin) an image using 4 points as controls.

RE:Map UV

The term UV mapping refers to the computer graphics terminology for defining an image that holds coordinates for texture mapping. RE:Map UV does just that. It looks up the UV map to apply a texture and properly filters out the projected result. Because the values in the UV image are spatial coordinates, it is important that no color remapping/conversion be applied to it before.

A UV map is a format most 3D systems can easily render images representing horizontal image coordinates in the red Channel and vertical coordinates in the green channel.



In this figure we have our background image on the top left, our UV map render on the top right. At the bottom we place two different source images to map onto the box as texture.

TIP: to preserve the shading (shadows…), simply render the object to map white (instead of green as in this example) and use a transfer function mode over the background like "Multiply" or "Overlay". You can as well always use the UV map alpha as a tracking matte to as well colorize (tint) the result.

Important Note: It is highly recommended you use this tool in 16 bpc. What happens in 8 bits is that you then only have 256 different values to represent the mapping on a surface which will create blocky results.

To use the plugin, you apply RE:Map UV to the UV map image sequence . In the Texture layer menu, you select the color source you want to use as a texture map. It will return a red frame until you set a Texture Map source. If you want to map only a segment of the Color Source (or if that source happens to have bad edges), you can go to "Warp Mode" Show Texture Map and define a crop area.

In the next section we define our UV image format for UV image maps. For now be advised that the color source can be a different resolution than the UV map itself. A black value in the UV map in the red channel will lookup in a pixel at the left edge of the Texture image. A full-on value in the UV map will look up a value at the right edge. Similarly, a black pixel in the green channel of the UV map will indicate a lower edge of the texture image and a full-on pixel in the UV map image will look up a pixel on the upper edge. A value of 50% red and 50% green will look up texture value at the center of the texture map.

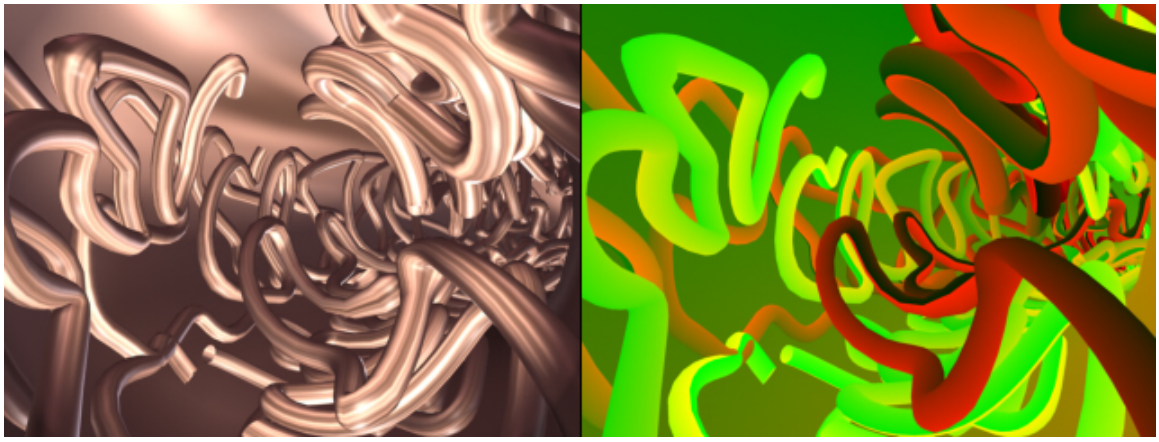
Basically the process is: Read the UV values of the main buffer/input, these will be values between 0 and 1.0 in X and in Y (the red and green channel). Take that value and index the color texture simply using (Uvalue * ColorImageWidth) and (Vvalue * ColorImageHeight). As such then two things are important to realize. A UV map is image ratio dependent and for proper anti-aliasing is resolution dependent (i.e. you should not resize the UV map). So, you are better to render the UV at the wanted resolution to start with. For example if you need to reuse the same material at NTSC and PAL res, it's better to render two UV map sequences. For finer results you may decide to render your UV maps at twice the resolution and simply scale the uv mapped image afterwards (after RE:Map) by 50%.

What is a UV Map?

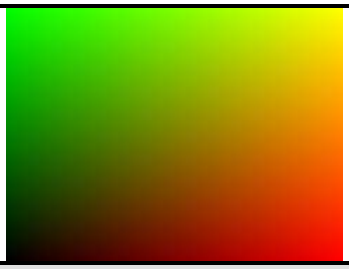
A UV map is an image that many 3D systems can render. It defines a normalized image coordinate system representing where a source texture image is projected. The unit UV map would be as illustrated below, from 0 to image-width is represented in the red channel and 0 to image-height is defined in 0 to 1 coordinates in the green channel. Note we we preserve alpha values when mapping using the UV map.
You can usually easily represent UV mapping with a simple script in a shading network. Maya users, make sure you get an alpha from the shading, so check in Custom Entities in the Render Settings the "Pass Current Alpha Channel" option. We recommend you use a shading process rather then a UV pass. In general note again if your renderer has a "UV" export mode like Max native renderer does, you still need to export a properly anti-aliased alpha channel of your object for the object edges to be properly rendered. If you can't figure it out in your renderer, to simply use an image as below with high-bit depth as texture map in your 3D system will work.

Identity UV Map. If your 3D software use a different channel mapping other than U→red and V→green, then you will have to rewire the channels (eg Set Channels) prior to applying RE:Map UV

As well on the compositing side, be careful about "Footage Interpretation". Most 3D renderers will render sequences that will require you to interpret the footage as premultiplied. Failure to do so might produce some edge artifacts, as it would change the meaning of the values at the object edges. Other things to watch: The UV maps are mathematically linear and any color space conversion will bias the result and your color will not have correct mapping. For example in Nuke, you can press the RAW button in the Image Reader to avoid such conversion if your map are 16 bpc. Also you cannot use UV maps that have only 8 bits. Finally you are probably better with straight formats that don't apply some gamma conversion, log conversion and what not, or perhaps even avoid formats like EXR 16 bit that are less precise in the high values.

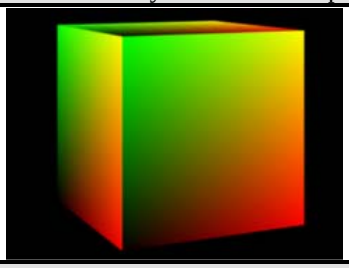


Tubes rendered and source texture map. In this case we also rendered the UV map for the background layer. You would get more precise edge handling and control by working in two passes (that is having the background pass be its own UV map layer). Source material provided by Tom Proctor.

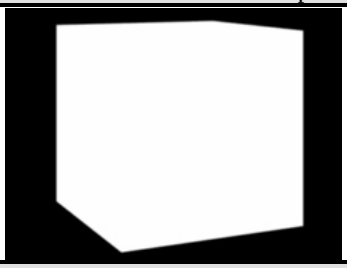


Identity UV texture map

Textured UV map



Rendered UV texture map of cube



Cube alpha channel (note antialiased edges)

	
Cube UV map rendered with texture	Another example texture image

RE:Map UV Controls

Warp Mode

There are 3 different Warp Modes for RE:Map UV.

Render

The output is the texture mapped using the uv map.

Texture Map

Allows you to look at the texture you want to map. This mode is to set a region of interest within the texture. You set the region of the source texture that will be using the crop points.

	
Show texture map mode with crop points set within Interior of texture.	Resultant UV textured image (Warp Mode set back to Render after cropping area is set).

Show UV with Edges

Allows you to view the edge tolerance used when creating the final render. This will be discussed below in more detail.

Check UV Map

This mode scans the UV map and checks to see if neighboring pixels have the same UV value, which can cause pixilation in the resulting texture map.

If a pixel has the same UV map value as one of its neighbors then a white pixel will be displayed, otherwise a black pixel will be displayed. White values are bad, so the goal is to have a final image that is all black. Many things can present problems in the UV map:

* Source UV map is in 8 bit, or the comp or project settings are 8 bit.
* Setting your shading rate in your 3D system so there is less then one ray per pixel (that would be min sample of -1 in Mental Ray)

* Using wrong texture map interpolation settings in your 3D system when using a source texture map with simple red and green ramps in the U and V directions.

Texture Map Source

The Texture Mapping (color source) we use to place on the UV map layer. It defaults to a full red frame to tell you you need to set something there. If you want to composite a UV map projected source onto another element (background) with some transparency level, simply adjust the alpha levels of the UV map image prior to applying this effect.

Crop Top Left TexMap and Crop Bottom Right TexMap: These two points allow you to define a cropping rectangle on the input texture. Note that if your UV map resolution and source texture resolution are different, you will see some squeezing or stretching of the texture map. That is because this viewing mode works in the UV map resolution and not the texture map resolution.. To set the crop area, simply set the "Warp Mode" menu to "Texture Map" and move the crop points appropriate. The cropping rectangle will be drawn in cyan.

Tile Repeat X and Y

"Tile Repeat" repeats the image into a number of tiles horizontally (X) and vertically (Y). Note that a value under 1.0 will be interpreted as using only a section of the UV map. Forexample, a value of 0.5 would mean that you only get half the texture, as opposed to setting it to 2 or 3, in which case you get the texture repeated 2 or 3 times).
A texture repeat of 2 in X and 2.5 in Y was used to make the picture above.

Flip

Allows you to flip the texture image you are mapping. Unfortunately some systems are lower-left origin and others upper-left origin, so to Flip Y is often necessary.

None

Has no effect

Flip X

Flips only the horizontal axis

Flip Y

Flips only the vertical axis

Flip XY

Flips both

Tile Repeat Pattern X and Y

Defines the way the texture rectangle is repeated horizontally and vertically

Repeat [[[[[[

Starts again once it arrives at the end of the UV mapping

Alternate [][][]

This mode is useful when you have abstract texture and want to create a seamless continuous effect

Cube UV map rendered with alternating repeat in Y direction.

Positon Offset X and Y

This value offsets the texture horizontally or vertically before the tile repeat is applied. Values should be in the range 0 to 1.0 (where 1.0 means shift by the whole image width (in X) or height (in Y)).

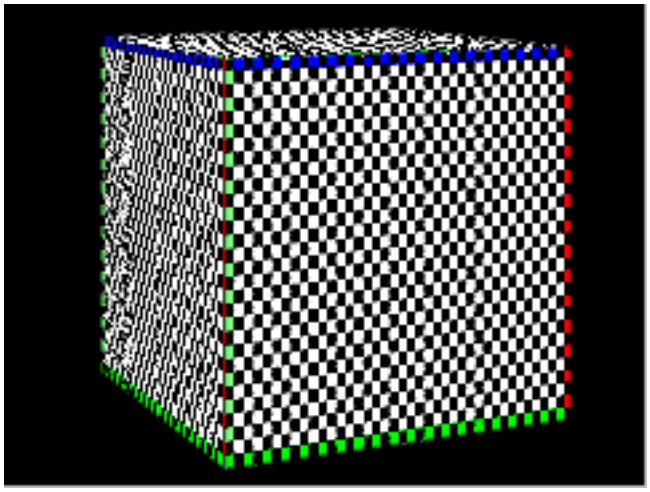
Offset of 0.25 in X used (moves image 25% in X).

Mip-Map

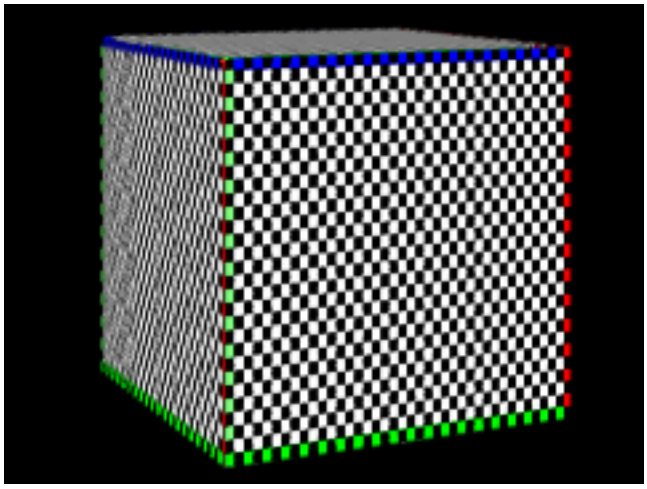
Turns on or off Mip Mapping. Mip Mapping is the texture filtering. It is off by default for quick interaction.

Mip-Map Amount %

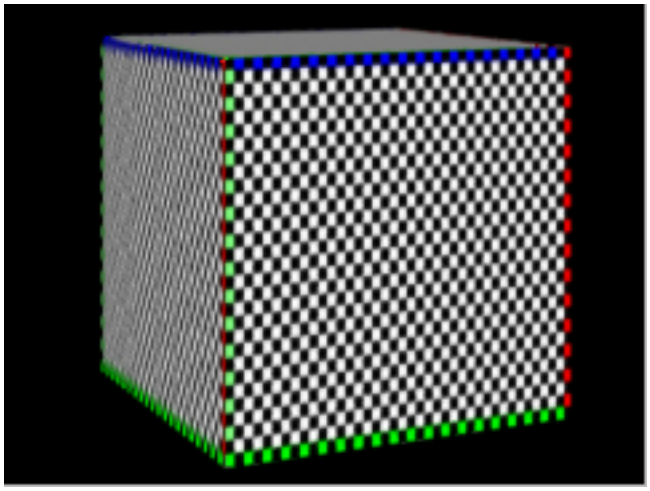
(range 0-100%). Sets the amount of filtering. The larger the value the more filtered (smoother) the result.



No mipmapping used.



Mipmapping used, 30%

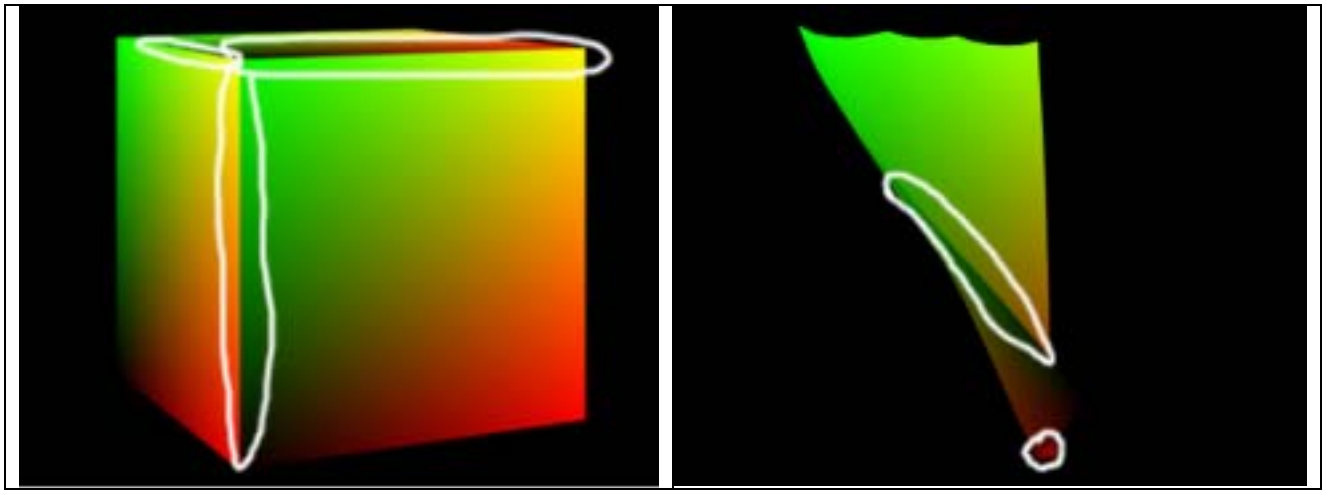


Mipmapping used, 60%

Edges Threshold %

When rendering a UV map, areas where different parts of the object meet, or cross another part of the object can often make for bad results because the pixels of the UV map have an antialiased edge where the two parts of the object come together. As such, the pixels along the edge in the UV map are a combination of two different places in the UV map, causing the resulting textured image to be incorrect.

Examples:

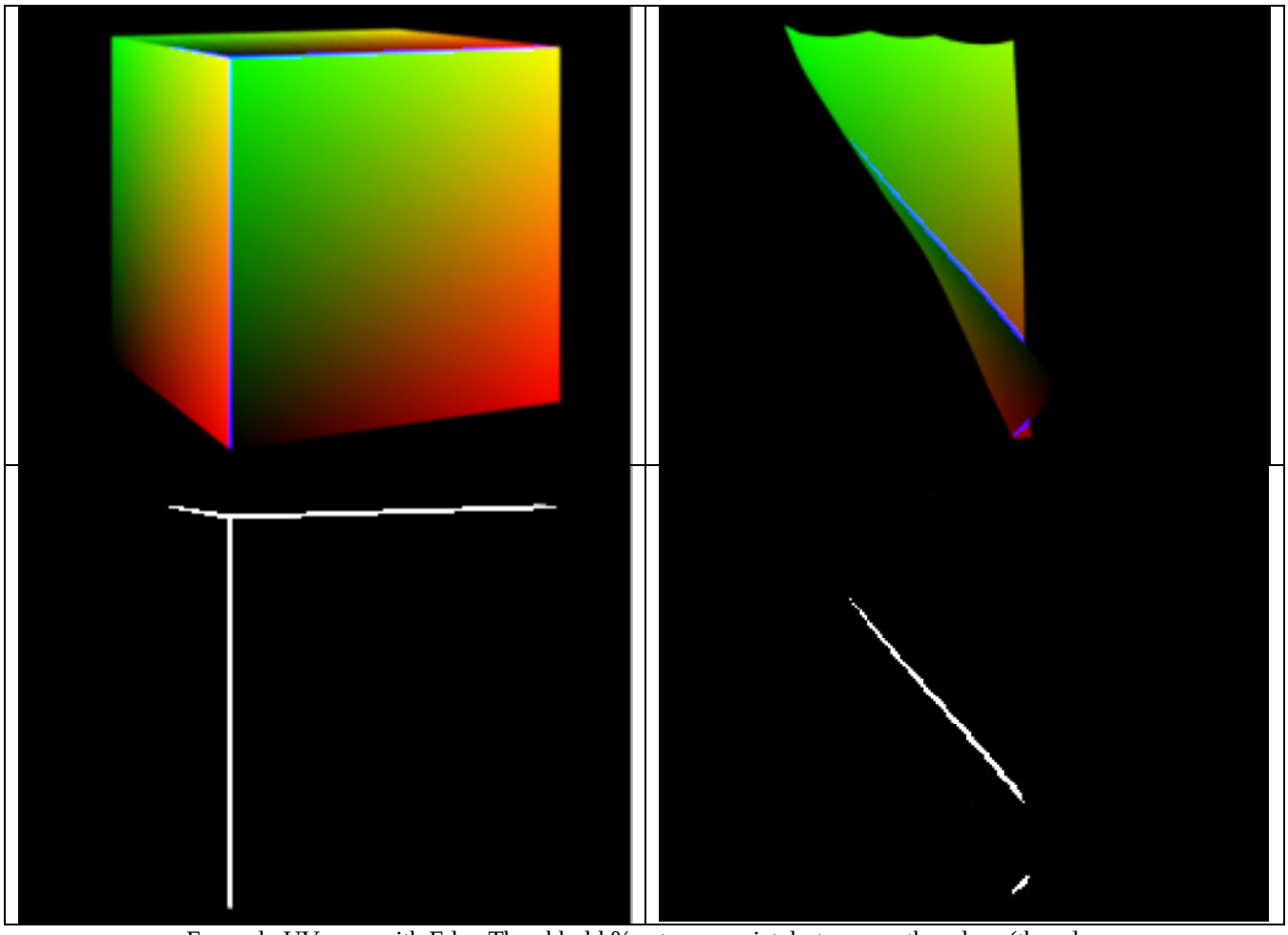


Cube and curtain UV maps with problematic areas of crossover areas or meeting edges highlighted.

Without any special handling of the edges, areas from the middle of the image are mapped to the edges in the texture map (notice dark pixel along the edges).

The plugin can do better if it knows where these edges of multiple-UV-averaged pixels are and can try to recreate the areas along the edge by using the UV information from both sides of the edge.

If you set the Warp Mode to Show UV with Edges you will see the UV map in the red and green channels. The blue channel will show where the plugin thinks the edges are in the UV map. By playing with the Edges Threshold % you can incorporate more or fewer pixels. Set this slider so that all multiple-value-UV pixels are covered with blue. Once you do this, you can set Warp Mode back to Texture and see the results.



Example UV map with Edge Threshhold % set appropriately to cover the edges (the edge threshold will vary each UV map). The lower set of images shows just the blue channel and is often helpful to view where the edges are actually set without the added visual UV values in the red and green channels

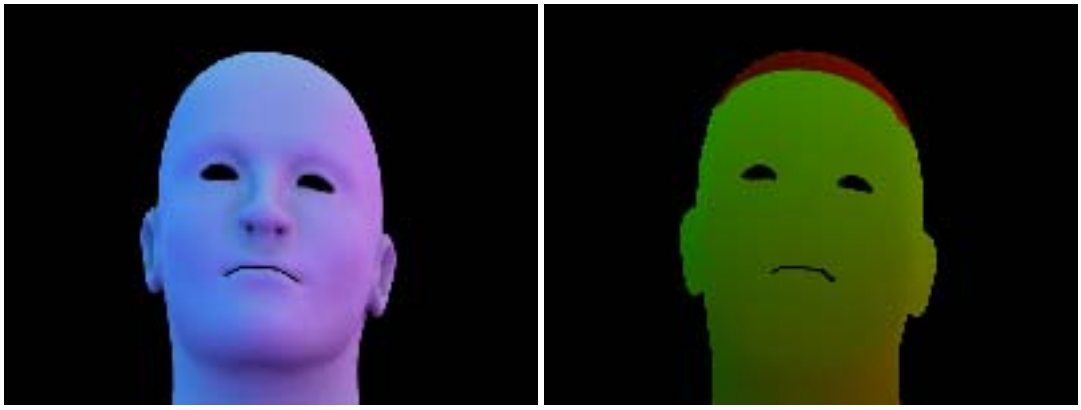
UV maps rendered with appropriately set Edge Threshold %. Note that the edges are appropriately textured, and no longer sample pixels of the texture map inappropriately.

RE:Map Inverse UV

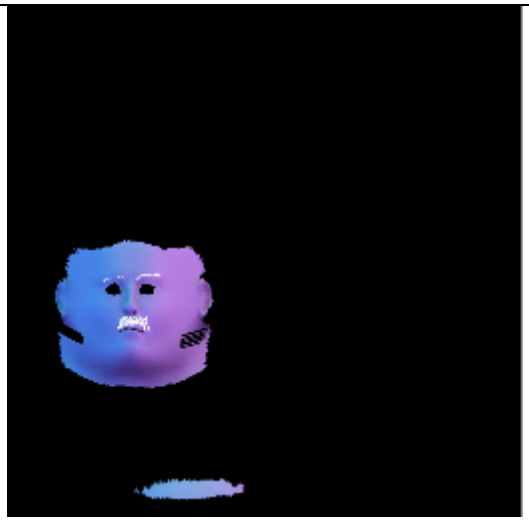
RE:Map Inverse UV takes as input a color image and a UV map image. The UV map is used to inversely project the color image to a rectangular image that may then be used to texture the UV Map with RE:Map UV, or for use as a texture map for your 3D system. Basically this plugin allows you to paint or otherwise modify a rendered image so that it can then be used to texture a UV map image sequence, basically performing the exact opposite operation that RE:MAP UV performs.

Since you might want to inverse to any arbitrary size, the main input is used to determine the output size. A simple way to do that is to use a simple constant image generator of the appropriate size. If you do so, remember that applications like Nuke then require you to set the In and Out time range so the effect is evaluated at every frame.

Let's say we wish to create a texture map for a 3D object and wish to work in a view from a particular camera viewpoint. Below are a UV map rendering of the object and a simple single-color rendering of the object used for reference.



What we'd like is a texture map that we can use to texture the object for the entire sequence. This plug-in allows you to paint and create a texture using the simple rendered image of the object as a spatial reference. Note that the top of the forehead represents a completely different area in the texture map (and is represented in the UV map by the dark red area at the top of the head).

	
Using the rendered object as a reference, eyebrows and mustache are created for the object	However, in order to render an entire animation with your 3D system (or with the previously described plugin RE:Map UV), an inversely project UV texture map can be created using RE:Map Inverse UV in order to texture the object for a whole animation.

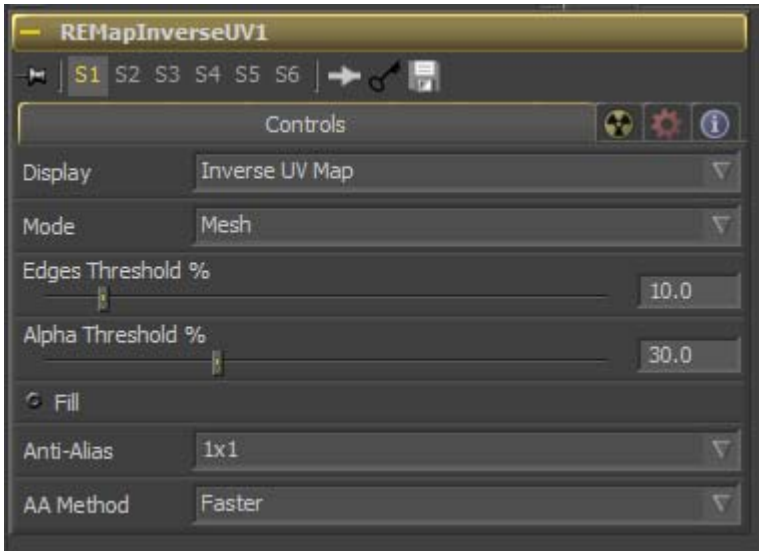
	
A more realistic source image is created by continuing to edit the original blue rendering. Here we see the final painted picture with the original render blended in at 50%	The final modified color image.

The final inverse texture map created using the more realistic image shown above.

Three notes:

1) Because the RE:Map Inverse UV preserves the alpha channel when inversely projecting, you don't always have to inverse project the original texture WITH the modifications you made or added onto it. You can certainly inverse project ONLY your changes of the original rendered image (the white eyebrows and mustache in the example above, but without the original blue-colored image underneath), inverse project using RE:Map Inverse UV and then reproject the texture map made for the entire animation using RE:Map UV, and then compositing the animation with just the changes made over the original sequence.
2) Cautionary note: if the original sequence is inversely projected at each frame using a different frame in time, then sometimes the texture map created will animate in an undesirable way. It is recommended to save out a single image and load it back in before use in RE:Map UV or with your 3D system (or set the comp where the inverse texture map is being created to a particular time, using the time remap features of the host application).
3) Sometimes there will not enough detail in a UV map to get enough detail in the resulting texture map, so it is often useful to continue adding changes to the inversely projected texture map AFTER it has been created. By doing this you can get a rough idea of where things land in the inversely projected texture map using RE:Map Inverse UV, and then continue to refine the texture map on the texture map after it has been inversely projected.

RE:Map Inverse UV Controls



Display

Inverse UV Map

This mode allows you to see the inversely projected texture map (as shown above).

UV with Edges, Thresh

This mode allows you to see where the plugin calculates edges (using the Edges Threshhold %) for use to calculate pixel connectiveness in the UV map, and the see the results of using the Alpha Threshhold % to discard pixels in the UV map with low alpha values. Both of these threshold percentage settings are described below.

As with the UV Map plugin, UV values at the edges of where the UV values wrap around, or where one part of the object crosses another, are problematic.

UV Map with calculated edges shown in blue. The Blue pixels are not used when inversely projected the texture map. A concrete example of the use of edges will be shown when discussing the Edge Threshhold % setting.

Check UV Map

This mode scans the UV map and checks to see if neighboring pixels have the same UV value, which can cause pixilation in the resulting texture map.

If a pixel has the same UV map value as one of its neighbors then ia white pixel will be displayed, otherwise a black pixel will be displayed. White values are bad, so the goal is to have a final image that is all black. Many things can present problems in the UV map:

* Source UV map is in 8 bit, or the comp or project settings are 8 bit.
* Setting your shading rate in your 3D system so there is less then one ray per pixel (that would be min sample of -1 in Mental Ray)
* Using wrong texture map interpolation settings in your 3D system when using a source texture map with simple red and green ramps in the U and V directions.

Note: if Check UV Map displays white pixels, and rerendering is not an option, you have the option of applying SmoothKit Diffusion or Gaussian (with Max Deviation setting of 10% or so, so that edges are not blurred over). It won't be as accurate as a rerender but can often make the UV map behave better.

UV Map

This is the UV map used to inversely project the Color Src. As with RE:Map UV, this map should be 16-bits per pixel or floating point. In the case we are showing here, the following picture is used as the UV map:

UV map for our example.

Color Src

This is the color map used to inversely project using the UV Map. **The Color Source must be the same resolution as the UV Map.**

Color map, which was made to register with the UV map.

Mode

Mesh

This mode connects inversely projected points with triangles, so that adjacent pixels in the UV Map get filled in when inversely projected into the calculated texture map.

Inversely projected texture, triangles used to render adjacent pixels in source color image.

Point Sample

This mode only inversely projects the UV samples into the calculated texture map, often leaving holes in the resulting calculated texture map. This mode is often useful when adjacent pixels, when filled in with triangles, cause the resulting texture map to be unrecognizable or unusable.

Inversely projected texture, point sampling used to render adjacent pixels in source color image. This mode will normally leave holes in the result but can be very useful when mesh rendering makes it difficult to figure out problems with the inverse projection.

Edge Threshold

This setting allows you to set an edge threshold to determine whether adjacent pixels should be considered connected or not (as is explained in the RE:Map UV plugin). The is particularly useful when parts of an object cross, or discontinuities in the UV map are encountered.

	
Edge Threshold set so that no edges are found in the texture map.	Resultant inversely projected texture map, which erroneously connects the top of the head to the forehead even though the UV map has a discontinuity there.

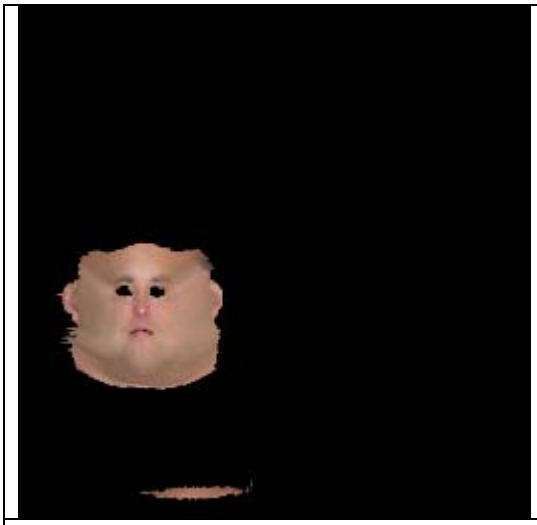
	
Edge Threshold set high so that no the edge on the top of the forehead is found.	Resultant inversely projected texture map, which disconnects the top of the head to the forehead by not connected pixels that were found to be edges in the UV map

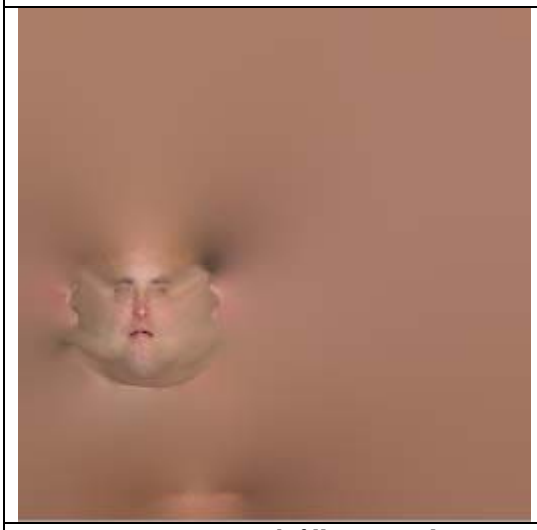
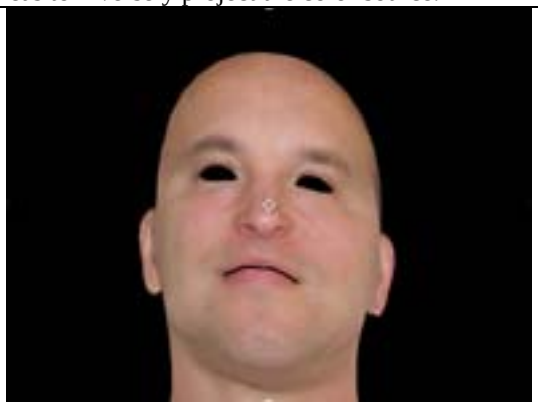
Alpha Threshold

This setting allows you to not use UV values in the UV map if the alpha is below a certain threshold. This is useful at the edges of objects where the UV value may not be of great use because its corresponding alpha value is very small.

Fill

If there are any holes in the resulting inversely projected texture map, this option allows you to fill them in with values from the surrounding neighborhood of calculated pixels.

	
Inverse Texture map with no filling.	Resultant texture map is unsatisfactory due to the areas of the texture map at the edges of the object (such as the bottom of the neck and the inside of the

	eyes) and the area where the texture map wraps around (the place where the forehead and top of the head meet) because the plug-in is not using those areas to inversely project the color source.
	
Inverse texture map, with filling turned on.	This texture map gives much better results when using as a texture for the object.

Anti-Alias

This option let you refine the calculated inverse texture map by supersampling 1,2, or 3 times in each of the x and y directions for each pixel.

AA Method

The two methods of Faster and Less Memory produce identical results. However, when calculating large inverse texture maps it may be necessary to use the Less Memory option (which is slower) in order to have enough memory to calculate the result. This is particularly true when using an Anti-Alias setting of 3.

RE:Map Displace

RE:Map **Displace** takes as input a color image and a displacement map image. The displacement map image is used to warp the color image.

The RE:Map **Displace** effect requires a 2D displacement map to perform it's magic, that is an image encoding some amount of displacement per pixel. We use the term "displacement map" to mean the vectors used to warp the Color Source. You will find in this manual, a section which goes into details about ways to generate displacement maps. Also found there is the format that we expect of displacement maps. Suffice it to say that the red channel controls the x-displacement and the green channel controls the y-displacement (the brighter the red channel, the more a pixel is displaced towards the right and the darker the more it is displaced to the left, similarly for the green channel and y-displacement). Motion Vectors unlike UV maps are "signed".

This is a simple illustration. On the left we have an image to be displaced. In the middle right we have a ramp in red (from bottom to top) and constant green (that, in this case, represents 0 displacement in y). This map we use as our displacement map. The right image is the result of applying the displacement map, which applies the progressively increasing x-displacement as we go from bottom to top, and 0 displacement in y.

Advanced example: Here the source image (on the left) is warped to the image in the center using motion vectors output by Twixtor PRO as shown on the right. The Warping factor here is 3.0 which probably extreme, but is used for the sake of. This is something you might want to do if you are adding 1 frame at the end or the beginning of a sequence that doesn't exist in the original sequence and want to extrapolate to create the new image instead of just duplicating the original end-frame.

Displacement Maps

Displacement maps often come from a 3D system (movement between frames of an animation), or from pixel movement analysis software, such as is found in our Twixtor product.

You can also make displacement maps manually with color gradient tools, render different forms of displacement from a 3D system (projected displacement, surface gradients), you can make them with any tools once you understand the mapping. You may also try using images found in your sequence already, although using those regular images as displacement maps often lead to confusing or undesired results.

For the geek in you: To get a directional gradient-based map, you can also use our Shade/Shape, which can calculate normals as output. The normals output mode of Shade/Shape can then be used as a displacement map for RE:Map Displace (perhaps in combination with a modulation channel to get the desired result).

Displacement map format
We support exactly the same format for displacement maps as with our other products such as ReelSmart Motion Blur and Twixtor. More information can be found here:
http://www.revisionfx.com/generalfaqsMotionVectors.htm . At this link we discuss motion vectors, but know that displacement maps are equivalent to motion vectors, as is our format for them.

Essentially we encode X and Y displacements in the red and green channel respectively and we use the alpha channel to indicate if an object is fully covering the pixel or not. Please note that the proper alpha information in the displacement image is important, and more details can be found here: http://www.revisionfx.com/generalfaqsMVFormat.htm . Scroll down the page to see the description of the usage of alpha.

Quick overview

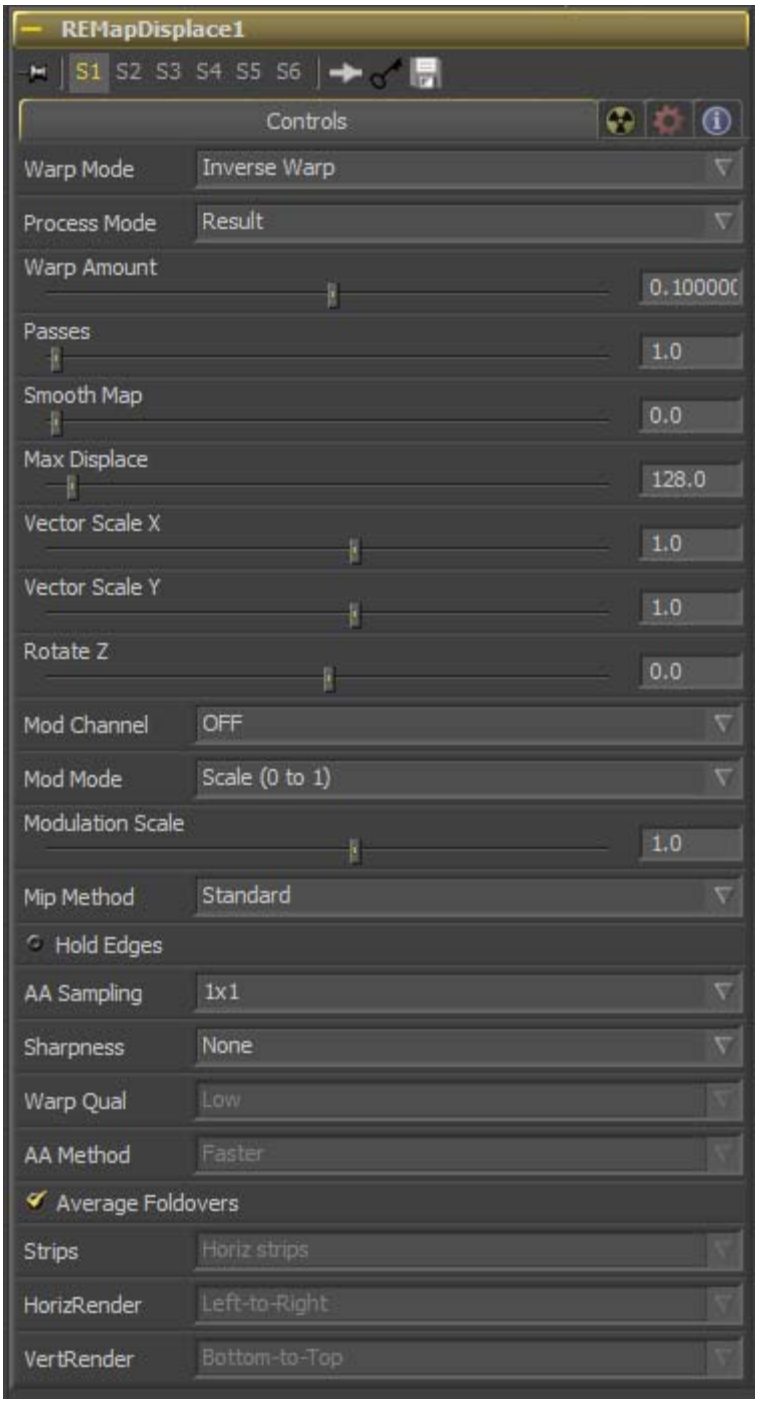
We assume that X is positive going from left to right. And the positive Y designates motion going UP (which is the opposite of Fusion coordinate system, for example). We assume that X,Y vector information is encoded in the red and green channels of the image. In addition we assume the vector information has been normalized so that both X and Y range from -1 to 1, presumably using some constant value to divide the X and Y component values. We'll call this normalization value the Max Displace value.

In 16 bits per channel, we assume that -1 corresponds to pixel value of 0 and that +1 corresponds to 65534. We have chosen to map (-1,+1) to (0,65534) because with this scheme we can represent a 0 displacement with a pixel value of 32767 (in a scheme that maps (-1,+1) to (0, 65535), a 0 displacement value corresponds to pixel value of 32767.5, which cannot be represented exactly).

```
// First, map -MaxDisplace to MaxDisplace to 0 to 1
float fred = ((x/MaxDisplace)+1)*0.5;

/* clamp values if needed */
if (fred<0) fred = 0;
if (fred>1) fred = 1;

/* assign pixel value */
unsigned short red = fred*65534.0 + 0.5; /* rounding is preferred to truncation, but that's your choice */

float fgreen = ((y/MaxDisplace)+1)*0.5;
if (fgreen<0) fgreen = 0;
if (fgreen>1) fgreen = 1;
unsigned short green = fgreen*65534.0+0.5.
```

RE:Map Displace Controls

Warping in RE:Map involves selecting a warping algorithm (Warp Mode), a warping amount and a number of iterations (Passes).



Warp Mode

The first menu options produces completely different effects.

Lets say we have the original image and displacement map.

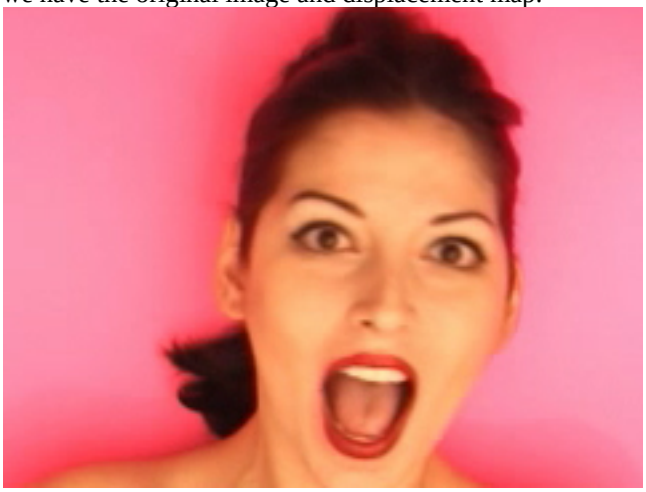
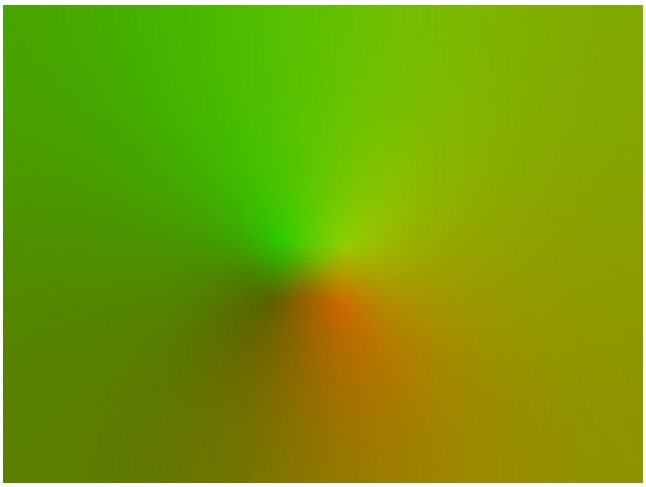


Image to be displaced



Displacement map which will magnify an area near the center of the image.

Forward

"Forward" mode performs forward warping. Forward mode is often most useful when you are using a displacement map generated from motion vectors analyzed from a sequence (e.g., using our Twixtor to generate vectors), or motion vectors from a 3D system. The Forward mode "puts" a pixel in a new location, based on the displacement value at that pixel.

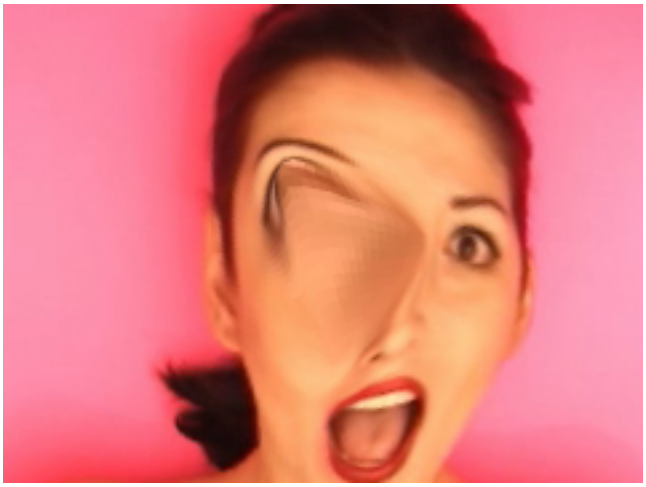


Image displaced using Forward Warp Mode.

Inverse

"Inverse" mode performs inverse warping. Note that the Forward mode "puts" a pixel and the Inverse mode "gets' a pixel. For small displacement amounts, the results may be very similar between the Forward and Inverse settings. The Inverse setting should probably be used when the Displace source is an arbitrary image or has very large displacement values and "forward" results are unpleasant. Unlike "Forward", "Inverse" does not create areas where the image can foldover onto itself, however it's mathematically an approximation and pushed too far will create a "liquid" like effect, similar to water ripples (which may be the desired effect).

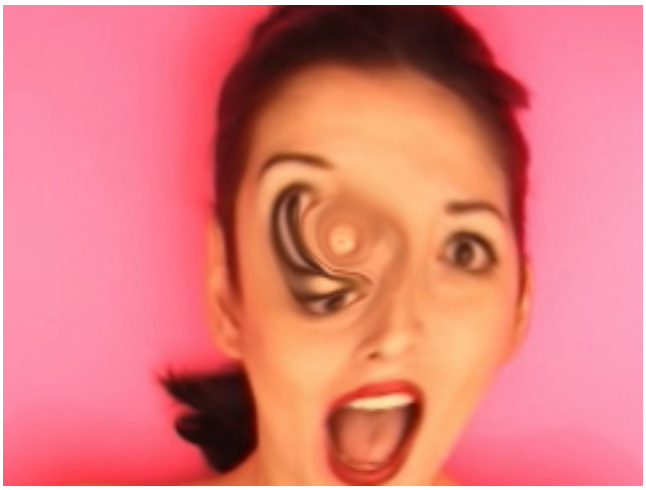


Image displaced using Inverse Warp Mode.

Inverse Warp mode is possibly very similar to other displacement map effect. However, in order to replicate something close to other displacement map filter, you'll need to set RE:Map Displace's Vector Scale X and Scale Y appropriately, which will be discussed below.

Smear

"Smear" moves pixels with the displacement amounts and embeds a directional blur between each pass (The Passes parameter will be discussed below). Net effect is to make everything extremely smooth.

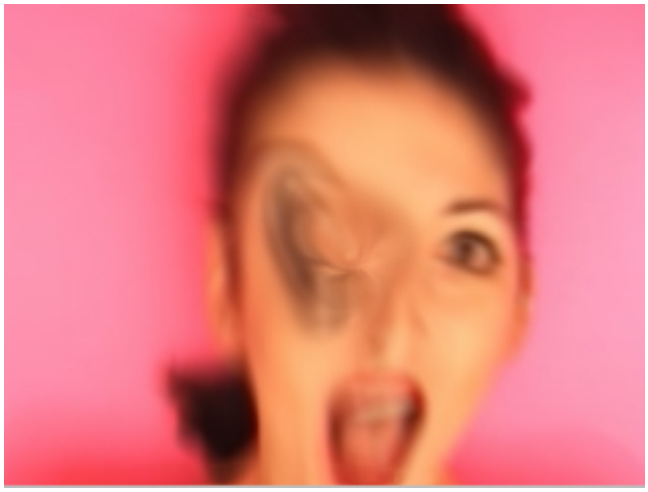


Image displaced using Smear Warp Mode.

Feedback

"Feedback" cumulates the displacement for the number of Passes set (if the number of Passes is less than 1, then feedback will not produce a significantly different result). Net effect: the feedback mode displaces the source image at different warping amounts (as set by the Passes parameter) and averages the displaced images. For example, if the Warp Amount is 1.0 and Passes is set to 6, then the image is warped multiple times by displacements of 1/6, 2/6, 3/6, 4/6, 5/6 and 6.6 of the original displacement map and then averaged together.

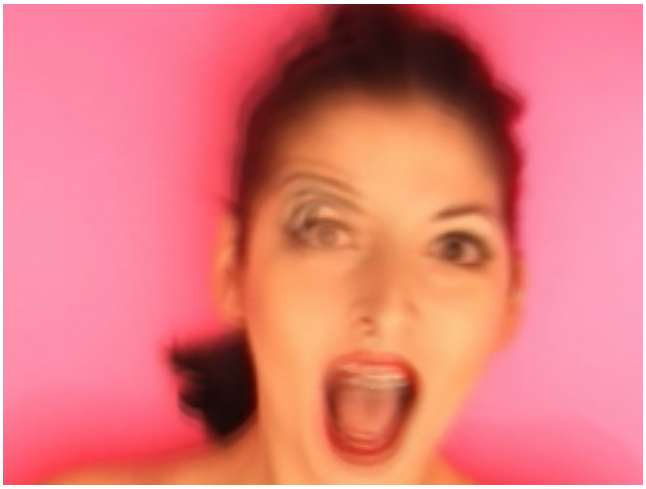


Image displaced using Feedback Warp Mode, with Passes set to 3.

Process Mode

Result

Displays the image warped using the displacement map.

Apply to Screen Coords

In this mode the plugin internally creates an "identity" UV map (a linear grayscale in Red and Green), and then ignores the Color Source, and instead warps that identity UV Map. The output can be used as a UV map elsewhere (for instance, in RE:Map UV).

Show Vectors

Allows you to see the effect of the Displacement Controls on the Displacement Map. The Displacement Controls include per-pixel scaling of the displacement, rotation, scaling, etc. and will be discussed later in the document. This Process Mode is a convenient way to see what vectors are actually being used internally.

Warp Amount

"Warp Amount" simply scales the displacement amount.
The Parameter Range is -10 to 10, that is 10.0 is ten times more displacement then 1.0 and 0.0 is no displacement, so should return the image unchanged. In some applications there is an interactive default range which we set to -1 to 1.0 so it behaves more nicely when you try to see what happens when you apply a different yet very close value, "right-click" on the parameter value will allow you to "Edit Value". In some applications you might not see the value when slide to increase the Amount, but you can type larger values.

Passes

The "Passes" parameter breaks the process into a number of passes by scaling the warping effect accordingly. Be aware that large number of passes might result in large rendering time. Simply, a value over 1 will iterate multiple times by breaking the process into the specified number of passes. This is useful particularly when using Forward Warping techniques where sometimes the image will foldover onto itself after mapping", slightly raising the "Passes" amount can such artifacts (at the expense of rendering speed) . Fractional amounts for Passes are allowed (so you may use 3.5 , for example). The Passes parameter is not meant to be animated. Once you have a look you like, use the Warp Amount slider to tune in and out the effect. MultiPass approaches will soften your images and might require you to set appropriately the Sharpness menu.

Displacement Controls

This group of controls contains more ways to achieve a desired result.

Displacement Source

With "RE:Map Displace", you need to specify a displacement source, the image encoding the displacement. To get proper precision, it is therefore suggested for better results to work in 16 bpc when using this plug-in. The Displacement Source can be of a different size than the color image. If they are of different resoutions then the Displacement Source is scaled spatially to match the color image source (which is the layer or clip that the plugin is applied to). Also depending on how the Displacement Source is generated, working with Displacement and Color Sources of different image resolution might require you to adapt the Scale X and Y parameters to get the desired results.

Smooth Map

(range 0-200) Smoothes the Displacement Map. Oftentimes a displacement can be noisy and produce a noisy results, it is usually useful to smooth a bit before using it as a displacement map. With this setting you tell the plugin how much to smooth the map before using it to displace the color source.

Max Displace

The maximum displacement used when you created thedisplacement vector image files. (see the discussion at the beginning of this document on the file format and the use of Max Dispalce).This group of controls is exactly as it is in ReelSmart Motion Blur and Twixtor if you are familiar with these products.

Scale X

(range -10,10): Individual Scale of the X displacement component once it has been internally converted to floating point.

Scale Y

(range -10,10): Individual Scale of the Y displacement component once it has been internally converted to floating point.

Rotate

(angle) “Rotate” turns the Displacement Map Vectors.

Modulation Controls

The Modulation Controls essentially scale the Displacement vectors on a per-pixel basis. These controls allow you to determine which channel of which image controls the per-pixel displacement scaling. This image channel must be the same image dimension (width and height) as the Displacement source. A full-on pixel in the Modulation channel tells the plugin to use the displacement at it’s full strength. A black (full-off) pixel tells the plugin to use no displacment. Greyscale values in the modulation image multiply the displacement used by the amount of the greyscale.

For example:

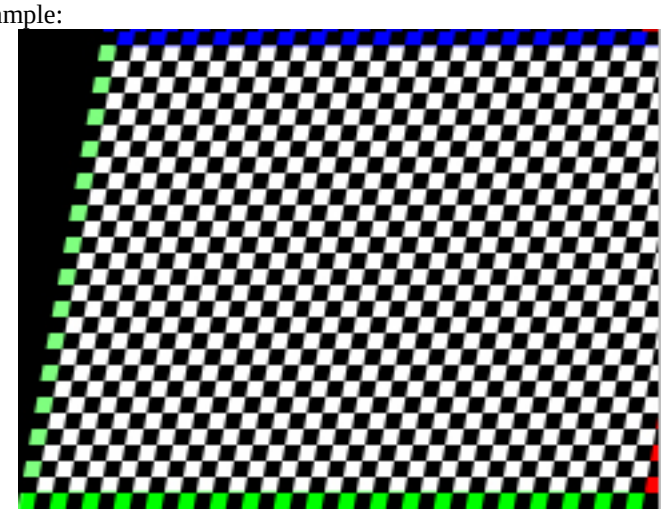
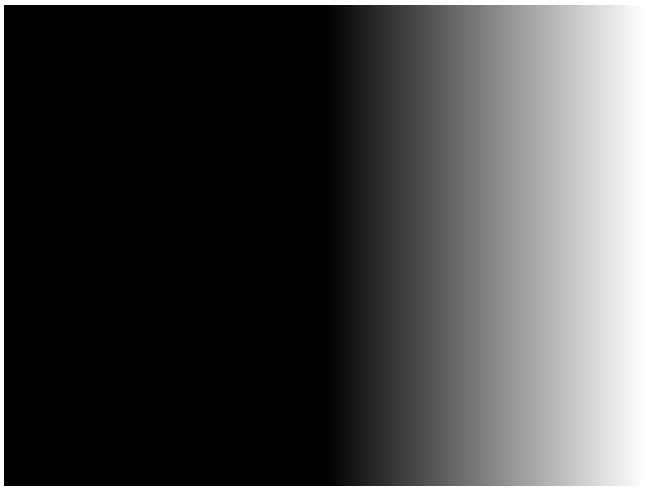
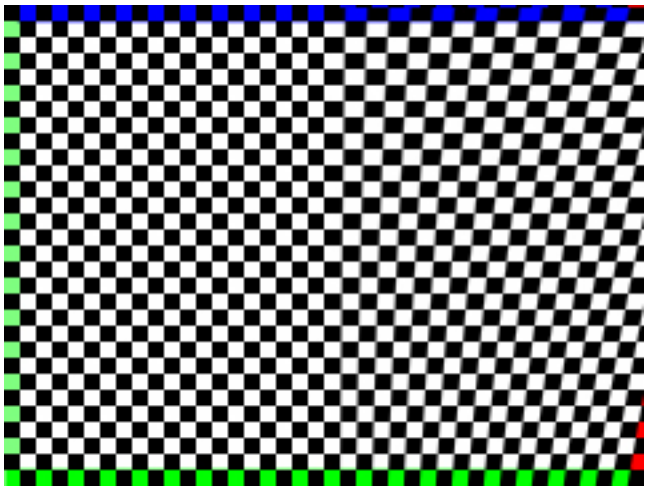


Image displaced with a displacement map.



Example modulation image. It is black on the left half and ramps to full-on in the right half.



Using the modulation image shown with the displacement map, the left half of the image gets no displacement (because there are black pixels in the left half of the modulation source), and the right half has the displacements slowly increasing to their full value for the right half.

Modulation Source

If None is specified, the Displacement Source is used as a modulation channel. **By default the Modulation channel is OFF (and not used).** (see next setting).

Modulation and Displacement Sources should be the same size. If you use a Modulation Source and its source is different than the displacement source, we will return a red frame to cue you that the resolution of the inputs do not match up.

Since we do not specify what is in the blue channel of displacement map sources, you are free to store a per-pixel scaling of the displacement in the blue channel of the Displacement Source if you wish.

Modulation Channel

By default the Modulation channel is OFF. You need to set this menu to a channel so the Modulation happens. Red, Green, Blue and Alpha here means channels 0 ,1, 2 and 3 of the image buffer.

Modulation Mode

There are two options. Scale 0 to 1.0 simply takes the grayscale image values of the Modulation Source Channel and use them as values 0.0 (full off pixel) to 1.0 (full on pixel) and shrinks (multiply) the displacement vectors by that amount. Scale -1 to 1.0, assumes a 50% grayscale value scales a displacement amount by 0 (so no displacement at that pixel), a fully black pixel is used to scale the displacement by -1.0 and a full-on pixel scales the displacement at the pixel by 1.0.

Modulation Scale

(range -10.0 to 10.0) This slider allows you to scale the Modulated Displacement. For example if a calculated per-pixel Modulation is calculated to be 0.5 and you set this slider to 5.0, then the internal value scaling value for the displacement at that pixel becomes a scale of 2.5.

Render Controls

MipMap Method

This option specifies how much filtering to perform on squashed areas of the image (areas that are smaller after displacement is applied). Note that the Medium and Smoothest settings will increase rendering time by a factor of 2 to 3.

Standard

This option performs a standard amount of filtering, but will often be enough for most purposes

Medium

This option performs a medium amount of smoothing on squashed areas

Smoother

This option performs a larger amount off smoothing on squashed areas.

Hold Edges

When "Hold Edges" is checked, it holds the edges of the image in place.

Anti-Aliasing Sampling

A menu option with 3 choices for over sampling: 1x1, 2x2, 3x3. For most purposes a setting of 1x1 will suffice. However, if you are creating areas of high curvature, or compressing areas of the image, you may need to set this higher. Artifacts will show up as stairstepping if the anti-aliasing setting is not set high enough.

Sharpness

Choose to either smooth or sharpen the image after displacing.
options: Smooth A lot, Smooth Medium, Smooth A bit, Ignore, Sharpen a bit, Sharpen Medium or A lot. The Smoothing is oriented toward killing sharp discontinuities (that meaning things like staircasing). The Sharpness is to recover some removed sharpness possibly as a result of multi-pass warping.

Forward Warp Controls

Forward Warp Quality

Quality setting is just that. Low, Medium, High. A Low setting will compute faster, but will not be as visually good. High is the best quality, and is naturally slower. Medium will suffice as the final rendering setting for most projects... This is only used when in "Forward" and "Feedback" Warp Modes.

Forward Warp Anti-Aliasing Method

There are two choices here and this option is only used in "Forward" and "Feedback" Warp Modes.

Faster

This method uses a big buffer to render the warped image before filtering down the result. That means for the 4 Anti-Aliasing choices the plug-in is using a buffer that is 1, 4, 9 or 16 times as big as the original image. Normally a 2048x1556 16bpc image is 25.5Mb (2048*1556*4 channels* 2 bytes per channel). For 4x4 over sampling the plug-in the plug-in would need 16 times this much memory, or about 408Mb).

Less Memory

This method internally uses a floating point buffer at the original resolution, but does multiple renderings slightly offset from each other in order to achieve the over sampling. Note that this method gives identical results as the Faster method, but takes more time. For the example given in the Faster discussion, the plug-in uses a buffer 2048 * 1556 * 4 (channels) * 4 (bytes for a float) , or approximately 51Mb instead of 408 Mb. Note that this method may often perform faster on larger resolution images if the previous method uses enough memory to cause combustion to use virtual memory.

Average Foldovers

When checked, this option renders folds in the warp so that each part of the fold can be seen (each part of the fold is "accumulated" in the result). If Average Foldovers is not checked, then pixels in the area of the fold will show the value of the pixel of the last time a pixel of the fold is rendered. As such the order of the render is important. To give you control over the order of the render, the following 3 settings are provided.

If Accumulate Folds is not checked, then pixels in the area of the fold will show the value of the pixel of the last time a pixel of the fold is rendered. As such the order of the render is important. To give you control over the order of the render, the following 3 settings are provided.

Strips

Vertical or Horizontal strips are rendered, as selected

Horiz Direction

Right to Left or Left to Right.

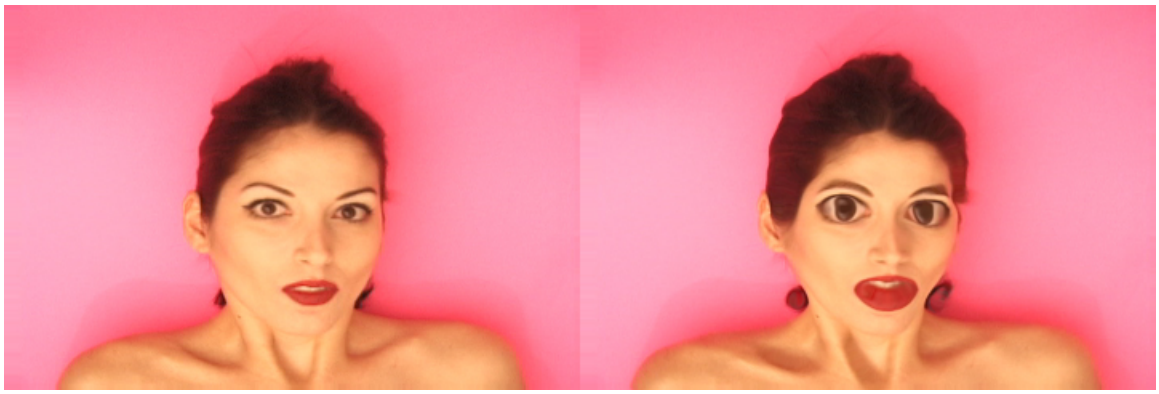
Vert Direction

Bottom-to-Top or Top-to-Bottom

RE:Map Distort

The “RE:Map Distort” effect allows you to automatically distort an image based on the features present, essentially creating a caricature of your video sequence by exaggerating the strength of the image features. It particularly works well on images with regions with smooth shading and hard features such as a human face. Unlike an effect like RE:Flex (a warping and morphing tool which allows you to perform precise warping by placing splines on the image), “RE:Map Distort” provides a quick way to distort images by playing with a few sliders.

Basically this plugin works similarly to RE:Map Displace, but generates the displacement map from the image itself, or another image (instead of you supplying an external displacement map directly).



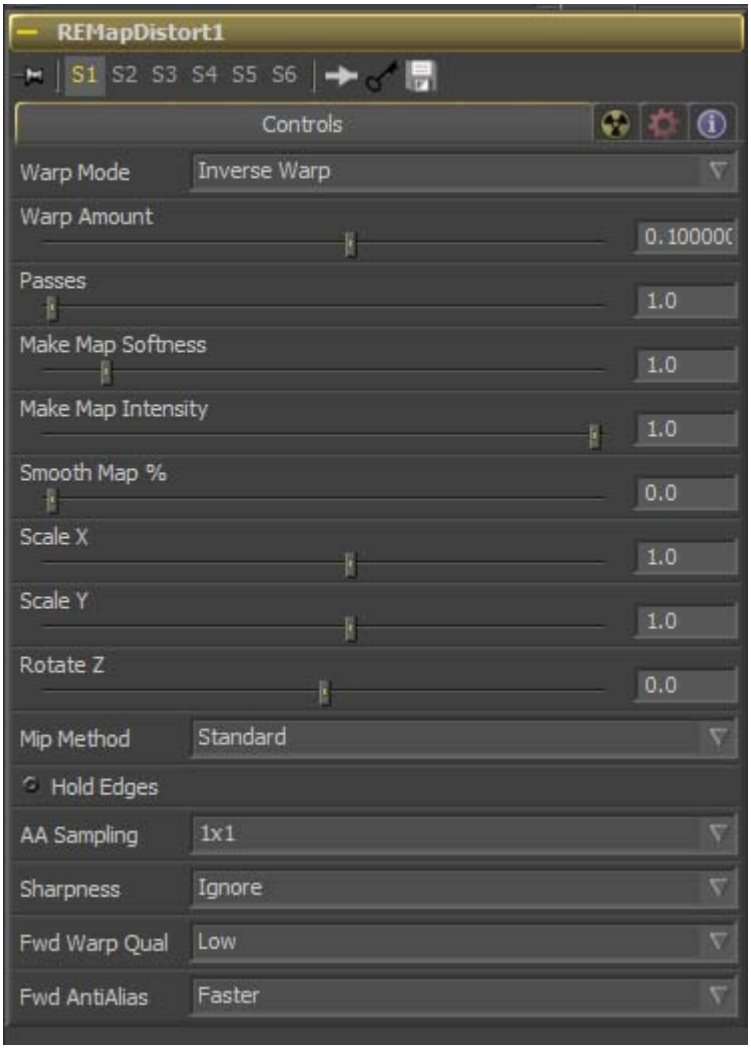
Source and some settings of RE:Map Distort. This test sequence was provided by Ami Sun, world@holophlo.net It is part of the tutorials that come with this product.

“RE:Map Distort” internally creates a direction image based on the features in the image, similar to ta per pixel Directional Blur”. The difference is that “RE:Map Distort” warps the image, instead of blurring it.

NOTE: We use below “Displace Source” to mean the color image from which Displacement is inferred. For this plugin, we use the term “Displacement Map” to mean the 2D Vectors derived from the Displacement Source used to warp the Color Source.

RE:Map Distort Controls

Warping in RE:Map involves selecting a Warping Algorithm (Warp Mode), a Warping Amount and a number of iterations (Passes). Then finer adjustments can be made playing with the Displacement Controls and the Render Quality settings.



Quick Start: Move the Warp Amount slider and see what happens.

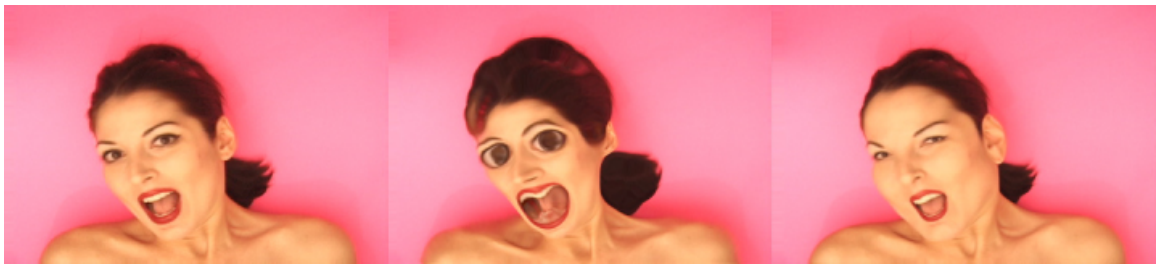
Warp Mode

The first menu option provides different warping settings. The four settings are Forward, Inverse, Smear and Feedback and work just like they do in the RE:Flex Displace plugin.

Warp Amount

"Warp Amount" simply scales the amount of displacement.
The Parameter Range is -10 to 10. 0.0 is no displacement, and will return the image unchanged. In some applications there is an interactive default range which we set to -1.o to 1.0 so it behaves more nicely. To expand the range "right-click" on the parameter value will allow you to "Edit Value" to expand the settable

range. If you want to dynamically turn on and off this effect, set the effect as you like on the largest "Warp Amount" value and then just animate the Warp Amount.



Warp Amount: 0.0, -0.4 and 0.2

Passes

The Passes parameter is exactly the same as is described in the RE:Map Displace plugin. The "Passes" parameter breaks the process into a number of passes by scaling the Warping effect accordingly. Be aware that large number of passes might result in large rendering time, particularly with the Smear Mode. Simply, a value over 1, will iterate multiple times by breaking the process into Passes. This is useful particularly when using Forward Warping techniques where sometimes small triangles can overlap, slightly raising the "Passes" amount will remove such artifacts (at the expense of rendering speed). The Passes parameter is not meant to be animated (it would create jumps in the animation if you did). Once you have a look you like, use the Warp Amount slider to tune in and out the effect. MultiPass approaches will soften your images and might require you to set appropriately the Sharpness menu.

Displacement Controls

This group of controls contains more ways to achieve the wanted result. The two controls, "Make Map" are something to play with if the effect is not exactly as you want, and the optimal setting varies for each image sequence.

ALT Displace Map Src

The ALT Displace Source is an option to use a different color source then the main input source in order to calculate the displacement map. Note that ALT Displace Map Src can be of a different size than the color image. If it is a different size, then the Displace Source is internally scaled to the Color Source resolution. It is not a good idea to use an ALT Displace Map Source that varies of size at every frame.

Source on the left, Result in the center and ALT Displace Map Src on the right. You can often get more controlled result by punching features (also maybe smoothing noisy stuff with something like SmoothKit Diffuse) . The image on the right displays an image used as an ALT Displace Source.

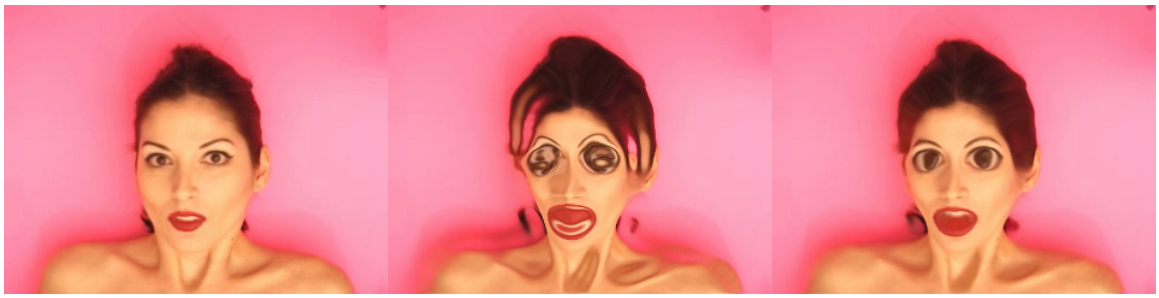
Make Map Softness

This parameter smoothes the Displace Source prior to calculating the Displace Map internally used by the effect, which spreads the influence of the effect out. For multi-pass effects (like Smear and Feedback), you often want to turn this setting up so the effect does not stick too much to the feature edges.

Original image on left. Default settings in the Center (Make Map Softness of 1.0, Make Map Intensity of 1.0 and Map Smoothness of 0.0) and on right “Make Map Smoothness” of 2.0.

Make Map Intensity

(range -1, 1) This parameter scales the intensity of the Displace Source, allowing you to change the brightness of the Displace Source image before the displacement map is calculated. In effect, this control allows you to reduce or increase the importance of edges before calculating the internal displacement map from the Displacment Color source image.

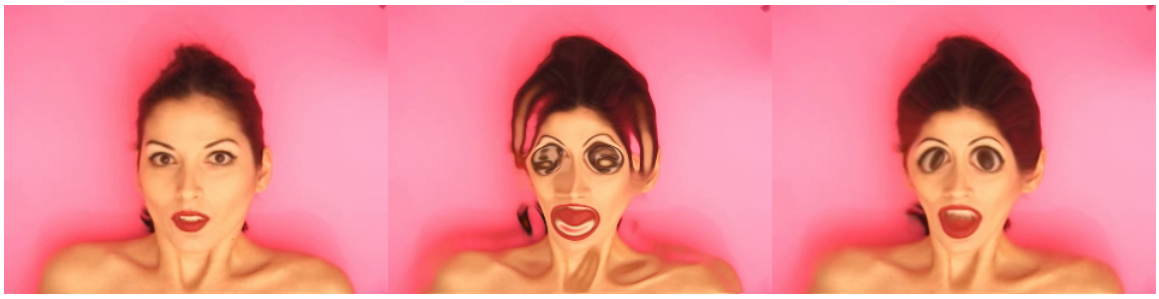


Original image on left. Default settings in the Center,. On the right “Make Map Intensity ” of 0.3.

Smooth Map

(range 0-100) Smoothes the Displacement Map after it is calculated from the Displace Color Source..Oftentimes the internal displacement derivation can produce slightly noisy results, so it is often useful to smooth the internal displacement values before the warp is accomplished.

.



Original image on left. Default settings in the Center and on the right “Smooth Map ” of 25.00.

Scale X

(range -10,10): Individual Scale of the X displacement component. If Scale X is 1.0 and Scale Y is 0.0 then it only warps horizontally.

Scale Y

(range -10,10): Individual Scale of the Y displacement component.

Rotate

(angle) “Rotate” turns the internal Displacement field. Note if all you want is to flip the direction you should simply use the Scale values for faster processing.

Render Controls

MipMap Method

This option specifies how much filtering to perform on squashed areas of the image. Note that the Medium and Smoothest settings will increase rendering time by a factor of 2 to 3.

Standard
This option performs a standard amount of filtering, but will often be enough for most purposes

Medium
This option performs a medium amount of smoothing on squashed areas

Smoother
This option performs a larger amount off smoothing on squashed areas.

Hold Edges

When “Hold Edges” is checked, it holds the edges of the image in place. This only has values if you don’t have a black margin around your frame. There are ways to pre-fill the margins, including “RE:Fill Frame Borders”.

Anti-Aliasing Sampling

A menu option with 3 choices for over sampling: 1x1, 2x2, 3x3. For most purposes a setting of 1x1 will suffice. However, if you are creating areas of high curvature, or compressing areas of the image, you may need to set this higher. Artifacts will show up as stair-stepping if the anti-aliasing setting is not set high enough.

Sharpness

Choose to either Smooth or Sharpen the Image.
options: Smooth A lot, Smooth Medium, Smooth A bit, Ignore, Sharpen a bit, Sharpen Medium or A lot. The Smoothing is oriented toward killing sharp discontinuities (staircasing). The Sharpness is to recover some gone sharpness for excessively smoothed results, probably as a result of multi-pass warping (setting the “Passes” parameter to some high value).

Forward Warp Quality

Quality setting is just that. Low, Medium, High. A Low setting will compute faster, but will not be as visually good. High is the best quality, and is naturally slower. Medium will suffice as the final rendering setting for most projects... This is only used when in “Forward” and “Feedback” Warp Modes.

Forward Warp Anti-Aliasing Method

There are two choices here and this option is only used in “Forward” and “Feedback” Warp Modes.

Faster

This method uses a big buffer to render the warped image before filtering down the result. That means for the 4 Anti-Aliasing choices the plug-in is using a buffer that is 1, 4, 9 or 16 times as big as the original image. Normally a 2048x1556 16bpc image is 25.5Mb (2048*1556*4 channels* 2 bytes per

channel). For 4x4 over sampling the plug-in the plug-in would need 16 times this much memory, or about 408Mb).

Less Memory

This method internally uses a floating point buffer at the original resolution, but does multiple renderings slightly offset from each other in order to achieve the over sampling. Note that this method gives identical results as the Faster method, but takes more time. For the example given in the Faster discussion, the plug-in uses a buffer 2048 * 1556 * 4 (channels) * 4 (bytes for a float) , or approximately 51Mb instead of 408 Mb. Note that this method may often perform faster on larger resolution images if the previous method uses enough memory to cause combustion to use virtual memory.

RE:Map Corner Pin

RE:Map CornerPin performs corner and inverse corner pinning.

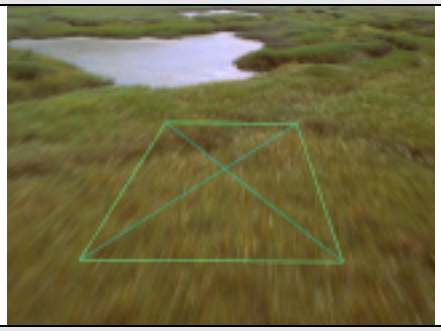
What is corner pinning? Planar mapping (corner pin) uses 4 points defining a quadrilateral to map. Imagine you had a rectangular layer in 3D space. The 4 corners of that layer are projected by the rendering viewing transform and now the 4 points have a position in screen coordinates. To directly set the 4 points in pixel coordinates will produce exactly the same result as a 3D plane projection if one can assume that the quadrilateral was a rectangle in 3D space. This allows us to perform what is often referred to as forward and inverse corner-pinning. We will use the expression here "quadrilateral" to mean 4 connected points in a 2D space and "rectangle" to mean an image definable in terms of position + Width, Height.

Since your desired output size can be arbitrary, use the main input (perhaps from a constant image generator) to set the node/effect output size.

IMPORTANT NOTE FOR NUKE USERS: Due to a bug in some version of Nuke 6 the following workaround is important to understand. We cannot initially automatically identify the size of the input. It will default to 0.,100 0,100 centered at 50,50. For that reason it is necessary for the user to press the **Reset** button to set the 5 points to their current initial location.

A simple example:

1. First we mark an area so it approximates the ground plane orientation.



2. Then we unproject so the texture is "front facing", using the four corner points of the quadrilateral, and draw a stroke.

3. Then we reproject the paint stroke.

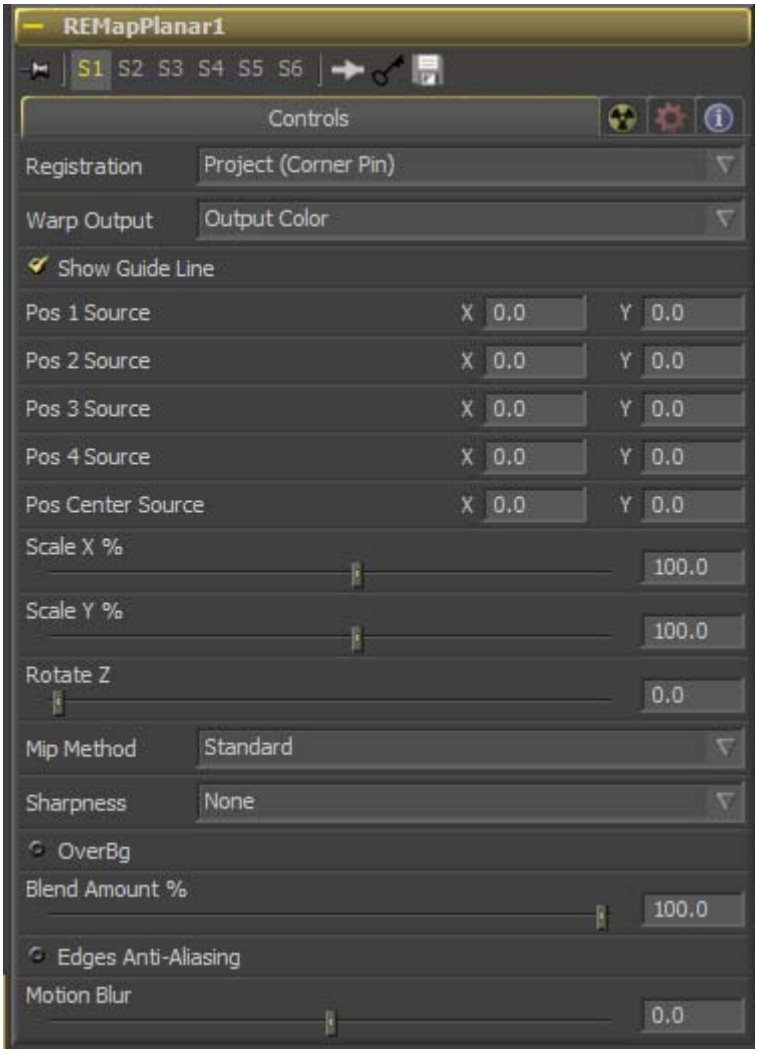
The source picture comes from Art Beats stock footage. As in this example, you sometimes need to sandwich the paint strokes between two instances of the RE:Map Corner Pin: one instance performs the inverse corner pinning, allowing you to work in an unprojected space, and one the other instance reapplies the corner pinning that was undone with the first instance. Please as well note that in some cases (particularly when the layer needs to be resized) you might need to pre-compose the first pass in order for this to work.

Layer
- Effects
  - RE:Map CornerPin 1 (to perform the inverse corner pinning).
  - Strokes
  - RE:Map CornerPin 2 (to reapply the corner pinning)

A bit of history

Such techniques are not really new. They predate the ubiquitous presence of computer graphics in our production techniques and particularly of 3D compositing environments. One of the first production companies to use such technique was Andre Perry Studios in the early 1980's. There Daniel Leduc (nowdays co-owner of an effects company called Hybride) developed a technique he called "ADO-tracking". The ADO was a device made by Ampex which allowed one to simulate rotating a video image in space. At that time as well, another device called the Bosch animation system came out, which was a simple real-time playback 3D system. One thing it lacked was texture-mapping. The technique developed by Leduc then was to take 4 points of a rectangular surface in 3D and use the ADO to essentially texture-map video in real-time in the online video suite. Corner pinning was also used in systems like Quantel to stabilize telecine material (when the film plane slides on the image plane, often when you have intermediates to work from with less perfs per frame). If anyone is interested by the actual math inside , the document http://www.cs.wisc.edu/~dyer/cs766/readings/heckbert-proj.pdf is an excerpt of Paul Heckbert original thesis on texture mapping and proof that a 3D rectangle projection can be represented uniquely as 4 points in 2D space.

RE: Map CornerPin Controls

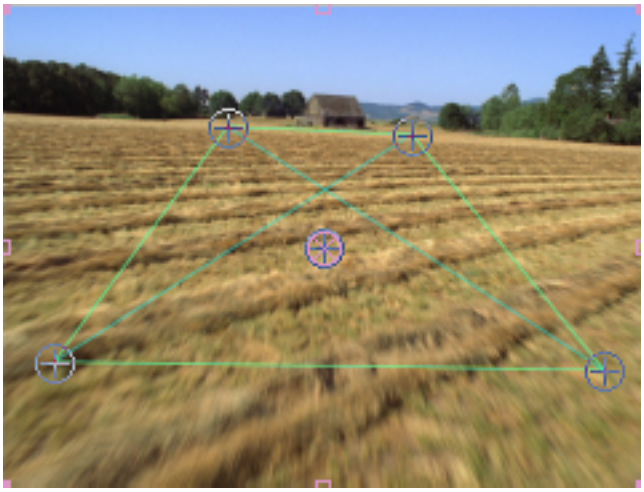


Registration Mode

You select the mode you want to work in. Normally you place points while in the " Source" mode.

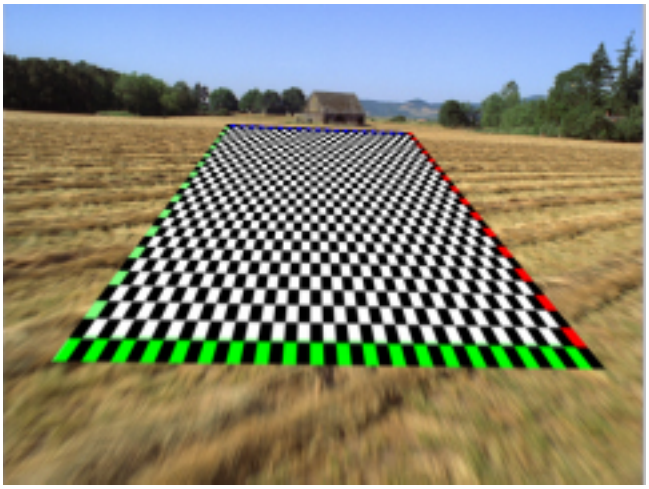
Source

Locate the 4 points on the actual source. (when in this mode the layer is not resized. Resizing options will be discussed below). Use this mode to register the 4 points for inverse corner pinning purposes.



Project (Corner Pin)

Projects the Source image into the Destination Rectangle formed by the 4 points location.



Another image is used as the “Alt Source” and is projected using the 4 points.

Unproject (Inverse Corner Pin)

Takes the 4 point positions and remove the projection, for example if you shot some text on a wall, you can make that element straight by unprojecting it.

Image defined by 4 corner points are inversely projected to remove the perspective distortion defined by the 4 points.

ALT Projection Source

If an Alt Source is provided, it will be the image "projected" instead of the image that the effect is applied to. Note we always show you the input source, not the ALT source when in "Source" mode.

Warp Output

The "Output Color" mode outputs the projected, unprojected or original color source as selected by the registration mode. In "Output UV" mode, we actually output UV coordinates that are used to index the source color image texture instead of the remapped color source image. Note "RE:Map UV" textures UV maps with supplied textures.

Overlays

This menus allows you to display the point overlay or not, and display the registration lines or not. The registration guide lines are burned in the image so should never be on when doing a final render.

Pos 1-4 Source

These are the 4 points defining our reference quadrilateral.

That said, with RE:Map CornerPin your comp, your current layer to which the effect is applied to, and the input layer parameter can all be of different dimensions.

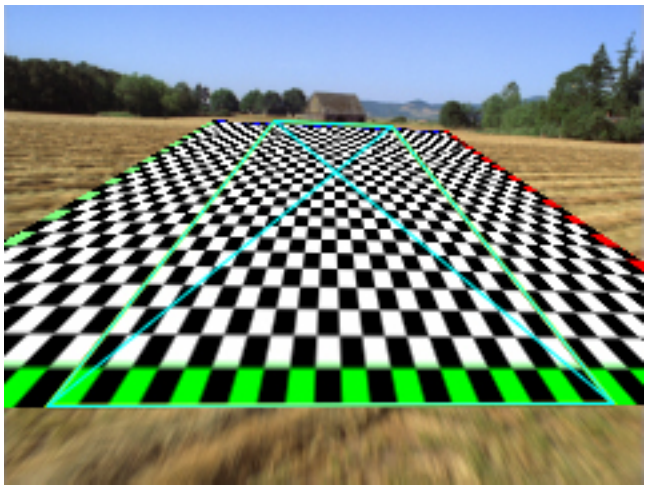
Transform Controls

Pos Center Source

This is the anchor point. It allows you to move around the image without affecting the projection.

Scale X and Y

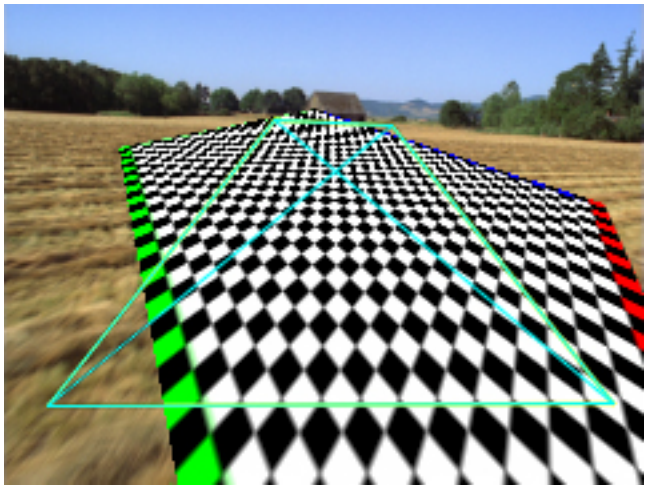
(Range -10000 to 10000). Scales the transformation, allowing you to project an image larger than the area defined by the 4 points (or conversely, shrink the area). This scaling happens in the projected space.



Result of an Scale X of 200. Note the rendered image goes past the quadrilateral defined by the 4 points.

Rotate Z

(angle) “Rotate” turns the Displacement Map.



Result of a Rotate Z of 45 degrees. Note that the rotation happens in the projected space.
and reenable the effect.

Render Controls

MipMap Method

This option specifies how much filtering to perform on squashed areas of the image. Note that the Medium and Smoothest settings will increase rendering time by a factor of 2 to 3

Standard

This option performs a standard amount of filtering, but will often be enough for most purposes.

Medium

This option performs a medium amount of smoothing on squashed areas

Smoother

This option performs a larger amount off smoothing on squashed areas.

Sharpness

Choose to either Smooth or Sharpen the Image after projection (or inverse projection).
options: Smooth A lot, Smooth Medium, Smooth A bit, Ignore, Sharpen a bit, Sharpen Medium or A lot. The Smoothing is oriented toward killing sharp discontinuities. The Sharpness is to recover some gone sharpness possibly as a result of multi-pass warping.

Over BG

If this box is checked, it will draw the projected plane over the background (the main input).

Blend % Amount

Simply use to define the amount of Blending to use if you render Over BG.

Edges Anti-Aliasing

This option grows slightly the layer and should be used only if you do very deformed planes and see some edge artifacts.

Motion Blur

Applies Motion Blur based on the actual transform (compared to previous frame). NOTE: Animating the Layer Grow X or Layer Grow Y amounts will produce an incorrect result for the motion blur. If you use a motion blur amount, please make sure your Layer Grow X and Y are not animated.

© 2009-2011, RE:Vision Effects, inc.