Tinder
for
Shake

Tinder for Shake User Guide

Rev: February 2010

Contents

INTRODUCTION

Welcome to this User Guide for Tinder on Shake. This subset of Tinder is designed to complement the rich set of tools and macros available to Shake compositors.

About this User Guide

This User Guide will tell you how to install, license and use the Tinder plug-ins. Each plug-in is described in detail in later chapters. This guide assumes you are familiar with Shake and the machine it is running on.

Installing Tinder

Tinder is available as a download from our web site www.genarts.com. Tinder is licensed using GenArts' licensing system. You will receive licensing information with your license purchase.

Installation and licensing instructions for Tinder are available on the GenArts website.

Documentation

Tinder comes with two sets of documentation, this pdf and a set of html files that connect to the Help button in Shake. This button will work immediately if you have installed to the default location, however, if you installed elsewhere the software must know where to look for the documentation. To do this, set the environment variable TINDER_DOCS_DIR to point directly at the docs directory. Multiple paths, separated by a colon, are not supported at this time.

About Tinder Plug-ins

All Tinder plug-ins integrate seamlessly into the Shake rendering tree. They are applied to your clips as you would any other node and they can all be animated using the standard animation tools.

Figure 1. Tinder panel.

COMMON PARAMETERS

There are many parameters that are common to some Tinder plug-ins. These are all described in detail in this chapter.

Filtering

Filtering is used to control the quality of your processed images by reducing the jagged lines characteristic of pixel devices. To render high quality images you should switch filtering on. With all image processing you have a trade off between quality and time. Filtering will increase the quality of your image but will also increase the time it takes to process the image.

Figure 2. Low Filtering

Figure 3. High Filtering

Parameters

filterQuality (string) **-** there are four options:

- **low -** highly distorted images may show jaggies. This is the fastest option.
- **medium -** this uses a Bilinear filter.
- **high -** this uses a MIP Bilinear filter. This is the slowest option.
- **globalDefault -** this uses the quality setting in Shake to determine the filtering method.

filterSharpness (float) **-** depending on the effect being filtered, the high filtering option may over-soften the image. To combat this, the filterSharpness control can be used to compensate. The default value of 100 is normal sharpness, increasing it will sharpen up the result, decreasing it will soften further.

Note *filterSharpness only has an effect when using the high filtering option.*

Blending

Many of the Tinder plug-ins have blending controls which allow you to specify how to mix between the image effect and its original source. Controls are also available to affect the gain of the image effect and its original source.

Figure 4. T_Etch with blend set to none

Figure 5. T_Etch with blend set to colour

Parameters

blendMethod (string) - sets how to blend the image effect with its original source.

- none - no blending is applied.

Note *Blend, Effect Gain and Source Gain have no affect when in this mode.*

- mix - displays a mix between the pixel colour values.
- add - displays a mix by adding the two images together.
- addOver -
- addUnder -
- screen - produces a bleaching effect. Light colours have more of an effect than dark colours. (A+B)-(A*B) or if you prefer 1-((1-A)*(1-B)) which is like combining the negatives of the two shots and "printing" the result.
- multiply - displays a mix by multiplying the two images together.
- difference - displays a mix by finding the difference between the two images. Dark colours will produce a more subtle effect than bright colours.
- subtractEffect - displays a mix by subtracting the image effect from the original source.
- subtractSrc -
- hardlight -
- darken - displays a mix by taking the darker of either the effect image or the original source.
- lighten - displays a mix by taking the brighter of either the effect image or the original source.
- EffectLum - displays a mix using the luminance of the original source and the chroma of the image effect.
- EffectChroma - displays a mix using the chroma of the original source and the luminance of the image effect.

- EffectHue - displays a mix using the hue of the original source and the saturation and value of the image effect.
- EffectSat - displays a mix using the saturation of the original source and the hue and value of the image effect.
- EffectValue - displays a mix using the value of the original source and the hue and saturation of the image effect.
- EffectColour - displays a mix using the hue and saturation of the original source and the value of the image effect.

Note *The Colour Blending method is the inverse of the Value Blending method.*

blendAmount (float) - sets the percentage mix between the image effect and its original source. A value of 100% will show just the image effect. A value of 0% will show just the original source.

Note *This control will have no affect if the Blending Method is set to None.*

Note *The inputs to this Blend are affected by the values set for Effect Gain and Source Gain.*

sourceGain (float) - sets the gain of the original source. The result of this is used in the Blend. The Source Gain also affects the alpha of the image. Setting Source Gain to 50% will half the brightness of the original source and will also half the value of its alpha. This will result in the underlying layers, or black if there are none, appearing through the original source.

Note *This control will have no affect if the Blending Method is set to None.*

effectGain (float) - sets the gain of the image effect. The result of this is used in the Blend. The Effect Gain also affects the alpha of the image. Setting Effect Gain to 50% will half the brightness of the effect image and will also half the value of its alpha. This will result in the underlying layers, or black if there are none, appearing through the effect image.

Note *This control will have no affect if the Blending Method is set to None.*

Lighting

Some plug-ins include controls for adding lighting effects. These allow the selection of a light position and brightness. Not all parameters are available for each plug-in. The complete lighting parameter list is included here.

Figure 6. T_Droplet lighting off

Figure 7. T_Droplet lighting on

lightType - controls the light model used.

- **local** - assumes the light source is a finite distance from the xy image plane as defined by the Distance parameter.
- **infinite** - assumes the light source is infinitely far from the xy image plane and will produce parallel light rays.
- **none** - switches off the lighting.

direction - controls where the light is coming from. It is defined to be the angle between the light and the horizontal x axis in the xy image plane.

elevation - controls the height of the light above the image plane. It is defined to be the angle between the light and an axis perpendicular to the image plane. Values between 0 and 180 will be visible. A value of 90 will set the light directly above the image plane. A value of 0 denotes a light source flush with the image plane.

Note *For many effects, light elevation values between 180 and 360 degrees will show no affect as this denotes a light source which is behind the image plane.*

distance - sets the distance the local light source is from the object.

strength - controls the brightness of the light source. The higher the value the brighter the light.

tightness - controls how quickly the light fades away with distance from the light source. Also known as fall-off. The higher the value, the tighter the highlights.

lightColour - sets the colour of the light source.

Source Edges

Some Tinder plug-ins need access to pixels which lie off the edge of the image. The edge of the image may be at the natural image boundary or pulled in with a crop or SetDOD node. The sourceEdges group controls which pixels are used in these calculations.

Note *In some plug-ins the sourceEdges group is called something else. The name will take the form of <input>Edges where <input> is the name of the first input.*

Parameters

sourceXEdgeMethod (string) - sets the behaviour of the image at its left and right boundaries.

- reflect - mirrors the image at the boundary.
- repeat - repeats the last line of pixels at the boundary.
- wrap - uses the pixels from the opposite edge.

- colour - uses a solid colour. This colour is set by the SetBGColor node.

sourceYEdgeMethod (string) - sets the behaviour of the image at its top and bottom boundaries.

- reflect - mirrors the image at the boundary.
- repeat - repeats the last line of pixels at the boundary.
- wrap - uses the pixels from the opposite edge.
- colour - uses a solid colour. This colour is set by the SetBGColor node.

Figure 8. Colour Edge Method

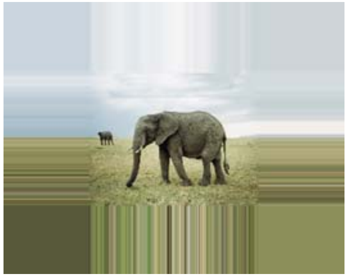

Figure 9. Repeat Edge Method

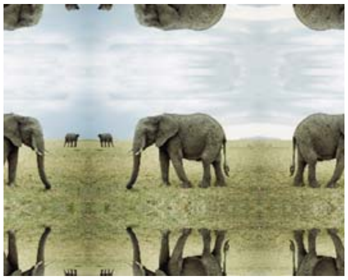

Figure 10. Reflect Edge Method

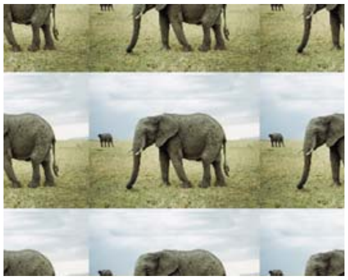

Figure 11. Wrap Edge Method

Attenuation Mattes

Some Tinder plug-ins have an optional second input for a matte. This matte is used to scale (attenuate) one or more of the parameters in the plug-in.

Where the matte is black the parameters will be multiplied by zero and the effect will not be seen in these areas of the matte. Where the matte is white the parameters will be multiplied by one, and the effect will be seen, Grey areas of the attenuation matte will scale the parameters accordingly.

TINDER

This chapter describes the plug-ins that are available in Tinder.

T_Caustic

Description

T_Caustic simulates the patterns created when light rays are reflected or refracted by a curved surface. Caustics can often be seen at the bottom of a swimming pool in bright sunlight.

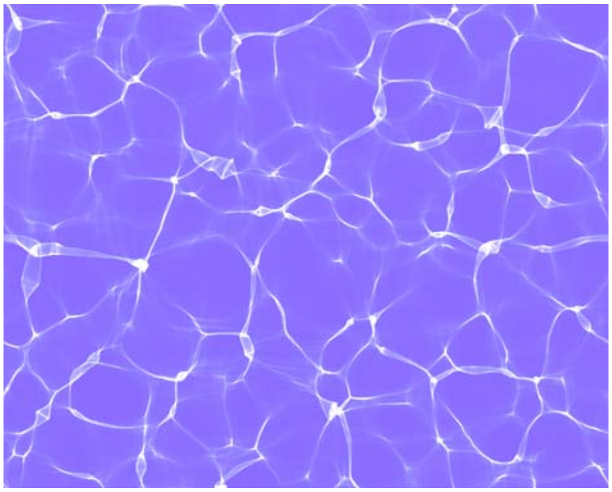

Figure 12. T_Caustic

Inputs

T_Caustic has one input - a source image.

Caustic

causticsWidth - the width of the rendered image.

causticsHeight - the height of the rendered image.

size - controls the scale of the image. Increase this to move closer to the water surface so that the caustics appear closer.

Note *Increasing the size may de focus the image. Increase the focus parameter to compensate.*

detail - controls the number of fractals used to generate the lines. Increase this for more detail and complexity in the lines.

brightness - controls the luminance of the lines. Increase this for brighter lines.

focus - controls the focussing of the lines used in the algorithm. Values close to zero will be out of focus or blurred. Increase this parameter to make the lines sharper.

speed - controls the rate at which the caustics move.

samples - controls the quality of the lines. Increase this parameter for smoother lines.

patternSeed - sets the random number used to generate the caustic pattern.

backgroundColour - sets the colour behind the lines.

foregroundColour - sets the colour of the caustic lines.

outputDepth - sets the bytes to render.

blending (group) **-** sets how to mix between the image effect and its original source. “Blending” on page 3

backgroundEdges -

version - states the version number.

Hints & Tips

To composite the caustics over the source image use the blending methods.

T_Distorto

Description

T_Distorto distorts an image using a matte. The amount of distortion corresponds to the brightness of the matte, and the distortion takes the form of a scale, translation or rotation of the image. T_Distorto

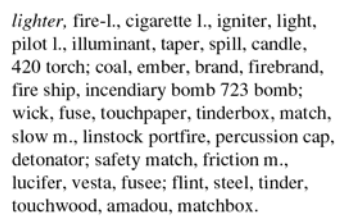
lighter, fire-l., cigarette l., igniter, light, pilot l., illuminant, taper, spill, candle, 420 torch; coal, ember, brand, firebrand, fire ship, incendiary bomb 723 bomb; wick, fuse, touchpaper, tinderbox, match, slow m., linstock portfire, percussion cap, detonator; safety match, friction m., lucifer, vesta, fusee; flint, steel, tinder, touchwood, amadou, matchbox.

Figure 13. Block of text

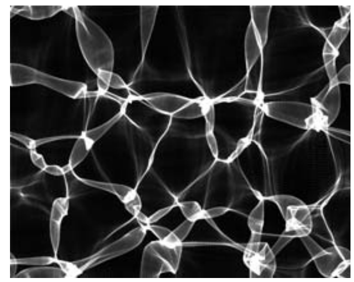

Figure 14. Water Caustics

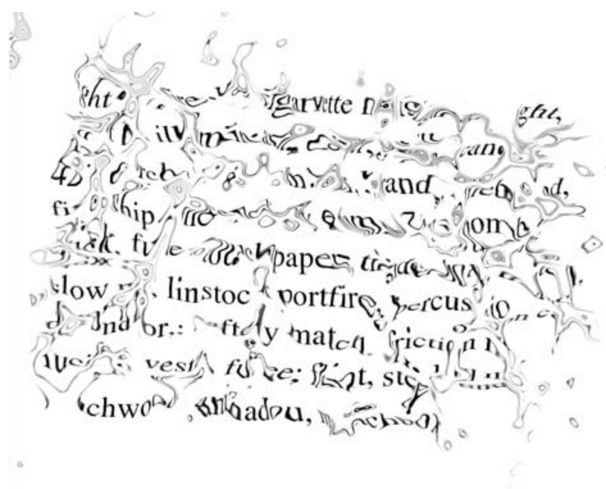

Figure 15. T_Distorto.

is often used to fake distortions seen through glass without having to render these in 3D.

Where the matte is black there is no distortion. Where the matte is white the full distortion is applied. A matte with smooth changes in luminance will gently ramp in the distortion to give fluid like effects.

Inputs

T_Distorto takes the following inputs.

- source (required)
- matte (required)

Parameters

strength (float) - is the global multiplier for the distortion parameters. Increase this to warp the picture more. A value of 0 produces no distortions. Use this parameter to ramp the effect in and out.

distortWith (int) - sets whether to use the luminance of the matte input or its alpha channel as the distortion source.
0 = Luminance
1 = Alpha

Source Distortion

+ distortionTransform (group) - open this to display the distortion parameters.

distortionOffsetX (float) - controls the horizontal deformation offset.

distortionOffsetY (float) - controls the vertical deformation offset.

distortionRotation (float) - controls the deformation rotation.

distortionScaleX (float) - controls the horizontal scale deformation.

distortionScaleY (float) - controls the vertical scale deformation.

transformOriginX (float) - controls the horizontal position of the source image.

transformOriginY (float) - controls the vertical position of the source image.

filtering (group) - sets the quality of the filter used when processing the effect. (See “Filtering” on page 2.)

+ sourceEdges (group) - this plug-in may need access to pixels which lie off the edge of the image. The parameters in this group control which pixels are used in these calculations. (See “Source Edges” on page 5.)

T_Fractal

Description

T_Fractal generates organic animating patterns.

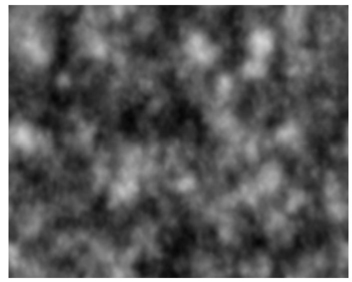

Figure 16. T_Fractal.

Inputs

T_Fractal takes the following input.

- source (required)

Parameters

outputDepth (int) - sets the output bit depth.

effectWidth (int) - sets the horizontal output size.

effectHeight (int) - sets the vertical output size.

tileWidth (int) - scales the fractal patterns in x.

tileHeight (int) - scales the fractal patterns in y.

spread (int) - controls the fractal density.

detail (int) - controls the fractal complexity.

levels (int) - controls the number of fractal layers.

speed (int) - sets the rate at which the fractals animate.

patternSeed (int) - changes the fractal pattern.

gain (int) - changes the brightness of the fractals.

backgroundColour (int)

foregroundColour (int)

+ blending (group) - sets how to mix between the image effect and its original source. (See “Blending” on page 3.)

T_Glass

Description

T_Glass gives the impression of viewing an image through a layer of distorting glass. Lighting effects are included.

Figure 17. Text used as the glass source.

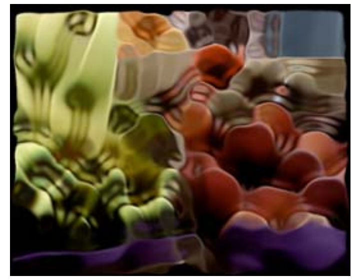

Figure 18. Fruit used as image and glass.

Inputs

T_Glass has three inputs - a source image, a mask for the glass distortion and an attenuation mask.

Glass

bumpScale - controls the amount the glass matte appears to deform the image. Positive values push out from the screen. Negative values indent.

distortionScale - controls the amount light rays are bent when entering the glass. This is also known as the refractive index of the glass. Changing this parameter alters the amount of distortion of the source image in the glass.

aberration - simulates defects in the glass that cause coloured fringing.

glassSoftness - controls the amount of blurring applied to the glass mask before the image deformation is calculated.

glassSoftnessAspect - controls the horizontal and vertical weighting of the blur on the glass.

imageSoftness - controls the amount of blurring applied to the image. This gives the effect of pulling the glass into focus and the entire background out of focus.

imageSoftnessAspect - controls the horizontal and vertical weighting of the blur on the image.

useClips - switch this on to active clipMin and clipMax.

ClipMin - pixels at or below this luminance value are set to black

ClipMax - pixels at or above this luminance value are set to white.

glassWith - sets whether to use the colour or the alpha of the second input as the distorting glass.

useGlassAsMask - switch this on to clip the effect with the glass mask.

lightColour - sets the colour of the light.

+ **lighting** (group) - controls the illumination of the image. (See “Lighting” on page 4.)

+ **filtering** (group) - controls the quality of your image. (See “Filtering” on page 2.)

+ blending (group) - sets how to mix between the image effect and its original source. (See “Blending” on page 3.)

+ **sourceEdges** (group) - this plug-in may need access to pixels which lie off the edge of the image. The parameters in this group control which pixels are used in these calculations. (See “Source Edges” on page 5.)

+ **glassEdges** (group) - this plug-in may need access to pixels which lie off the edge of the image. The parameters in this group control which pixels are used in these calculations. (See “Source Edges” on page 5.)


Figure 19. Stained glass window


Figure 20. As seen through T_Glass

T_LensBlur

Description

T_LensBlur simulates the true defocusing properties of a camera lens to give realistic focus pulls. It includes controls for the shape of the camera diaphragm, highlight blooming and chromatic aberration.

Figure 21. Cars at night.

Figure 22. Defocus with T_LensBlur.

Inputs

T_LensBlur takes the following inputs.

- source (required)
- matte (optional)

Parameters

xPixels (float) **-** controls the horizontal size of the blur.

yPixels (float) **-** controls the vertical size of the blur.

gain (float) **-** controls the brightness of the image.

aberration (float) **-** controls the defects in the lens that causes the image to display coloured fringes.

+ lens (group) **-** open this to display the lens parameters.

diaphragmShape (int) - sets the shape of the highlights. If polygonal, the number of sides is determined by the numberOfSides parameter.

- **circular**
- **polygonal**

numberOfSides (float) - sets the number of sides of the polygonal highlights. This is only active if the diaphragm shape is set to polygonal.

angle (float) **-** controls the rotation of the highlights. This is only apparant if the diaphragm shape is polygonal.

lensType (int) **-** sets the type of lens simulated.

- **normal -** the light is refracted through the lens before exposure on the film.
- **catadiatropic -** the light is reflected by mirrors before reaching the film. This method produces a dark circle at the centre of the highlight.

catadiatropicSize (float) **-** sets the size of the dark circle at the centre of the highlight.

scissorAngle (float) **-** controls the clipping of the highlights into another shape.

+ **bloom** (group) **-** open this to display the bloom parameters.

blooming (boolean) **-** toggle this on to simulate the over exposure of the effect.

bloomThreshold (float) **-** controls the luminance level above which pixels will bloom. Decrease this if no blooming is seen on the image.

bloomGain (float) **-** controls the brightness of the blooming.

bloomClamping (int) **-** controls the value of the pixels outside the legal range.

- **pixelMax -** the blooming will produce colours up to peak white.
- **imageMax -** the brightest pixels of the output image will not exceed the brightest pixels of the original image.

maskBlooming (int) **-** controls the use of the second input as an attenuation matte to control the blooming. The matte attenuates the threshold value between the current parameter value if the matte is white to a threshold value of 100 if the matte is black.

- luminance
- alpha
- negativeLuminance
- negativeAlpha

+ **sourceEdges** (group) **-** this plug-in may need access to pixels which lie off the edge of the image. The parameters in this group control which pixels are used in these calculations. (See “Source Edges” on page 5.)

T_LensFlare

Description

Camera lenses are designed to focus light onto a photo-sensitive surface. The lens housing often contains many individual glass lenses through which the light is refracted. However, a small percentage of light is reflected from the surface of the lenses and this reflected light forms the lens flare patterns we see on the photographed image. Lens flares are most noticeable when the lens is pointed towards a very bright light.

Figure 23. T_LensFlare showing faint red ring around the sun and the trail of polygons.

T_LensFlare generates realistic and highly customisable lens flares. It can be used over a background image or to generate an image from scratch. The lens flare is made up from four components:

1. **Rays**. These are formed from two elements - a bright soft glow and long light rays.The bright soft glows represent overexposure on the photosensitive surface as the camera points at the bright light source. There are four different types of rays in each of the ray components. These are shown in Figure 24. Spikes (top left) have unequal light rays, Rays (top right) have equal length light rays, Uniform Spikes (bottom left) have equal length light rays and are equally spaced round the glow, Diced Spikes (bottom right) have unequal length dashed light rays with no centre glow.

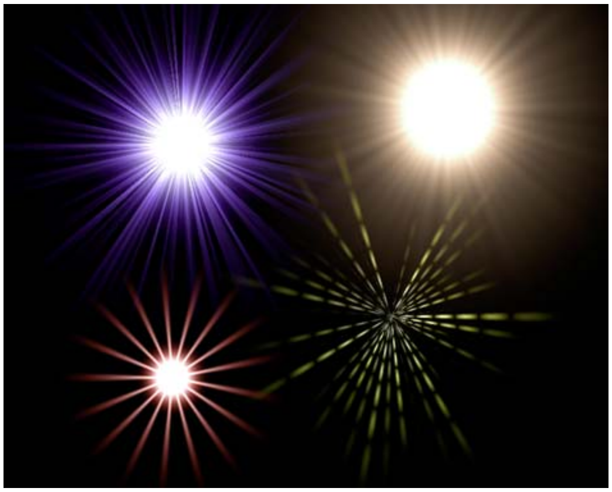

Figure 24. There are four different types of highlight. Clockwise, starting top left. Spikes, Rays, Uniform Spikes and Diced Spikes.

2. **Shards**. There are two bright horizontal light rays that are characteristic of lens flares from an anamorphic lens.

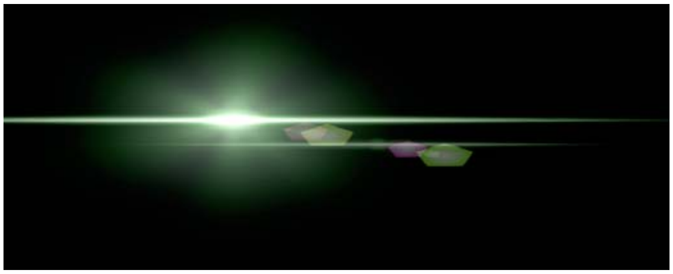

Figure 25. Light shards characteristic of a lens flare through an anamorphic lens.

3. **Polys**. These are the circular or polygonal coloured glows that lie on a line from the center of the rays through the pivot point. The shape of the polygons are defined by the shape of the iris that forms the camera aperture. The iris is built from a series of interconnected metal blades. If the aperture is opened wide the blades form a circle and when stopped down they form a polygon.

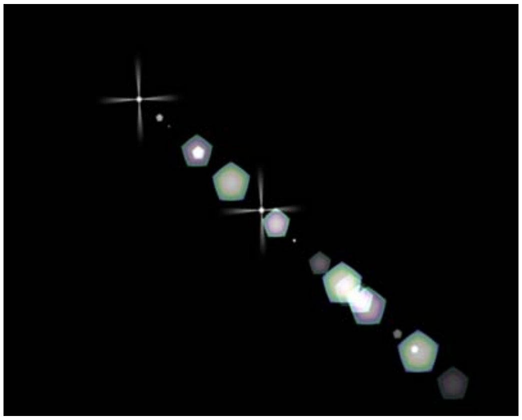

Figure 26. Polygonal artifacts. Shown also in white is the ray centre position (top left) and the pivot position (centre)

4. **Rings.** There are three rings which form around the highlight.

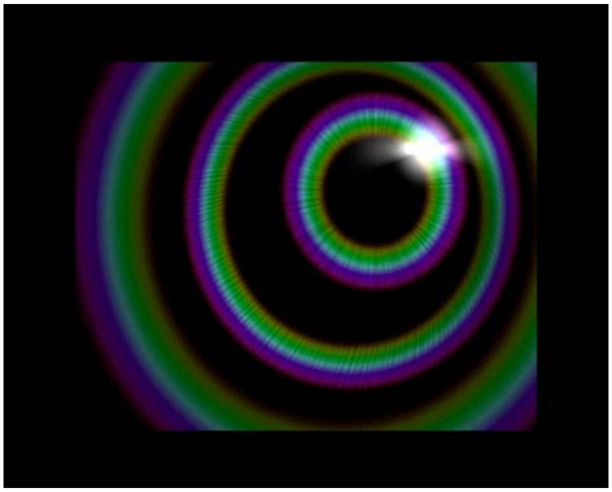

Figure 27. The three chromatic ring styles. From inner to outer is Chromatic Ring, Chromatic Radial and Chromatic Ripple. The highlight position is shown as a bright white spot top right. Note the dark intersections of the middle Chromatic Radial emanate from the highlight position and not the centre of the ring.

The rings can be rendered in one of five styles: Halo Ring, Chromatic Ring, Chromatic Radial, Chromatic Ripple and Arc.

Inputs

T_LensFlare takes the following inputs.

- background (optional)
- obscureMask (optional)

Parameters

outputDepth (int) - sets the colour depth for the output image.

0 = Auto
1 = 8 bit
2 = 16 bit
3 = float

xCenter (float) - controls the horizontal position of the center of the rays.

yCenter (float) - controls the vertical position of the center of the rays.

xPivot (float) - controls the horizontal position of the pivot point for the polygons.

yPivot (float) - controls the vertical position of the pivot point for the polygons.

flareAspect (float) - controls the horizontal and vertical weighting of the effect.

gain (float) - controls the overall brightness of the lens flare. A value of zero switches off the lens flare. This value can also be controlled using the second input obscuration mask.

seed (float) - some parameters use a random number in their calculations. The seed value changes the random number sequence giving a subtly different look to the lens flare.

obscuring (int) - sets whether (and how) to use the second input to attenuate the overall gain.

0 = None
1 = onBlack
2 = onWhite
3 = onTransparent
4 = onOpaque

obscureSize () - blurs the obscuration mask so that the gain ramps in and out at matte edge boundaries. A value of zero will ensure that the lens flare switches on and off immediately on crossing a hard matte boundary.

Rays & Glows

+ rays - open this to display the ray parameters.

rayForm (int) **-** controls the type of rays. See Figure 24 on page 17.

0 = none - switches off the rays.
1 = rays - renders light rays of equal length and unequal spacing around the centre.
2 = spiked - renders light rays of unequal length and unequal spacing around the centre.
3 = uniformSpikes - renders light rays of equal length and spacing around the centre.
4 = dicedSpikes - same as Spiked but with the luminance variations along each light ray giving a dashed appearance. These dashes are controlled by the rayCoreSize parameter.

rayOffset (float) **-** controls the shift in position of the rays from the centre to pivot.

rayRotation (float) **-** controls the rotation of the rays around the centre.

raySize (float) **-** controls the length of the rays.

rayCoreSize (float) **-** controls the size of the bright glow at the centre of the rays. If the rayForm is set to Diced Spikes the glow disappears and is used to control the luma variations along the length of each ray. Reduce the rayCoreSize to get more variations.

raySpokeWidth (float) **-** sets the width of the rays.

raySpokeCount (float) **-** sets the number of rays drawn from the centre.

rayFractalDepth (float) **-** controls the amount of random perturbations applied to the spokes to make them more interesting and more numerous.

rayGain (float) **-** controls the brightness of the rays.

rayColourCore (colour) **-** sets the colour of the centre glow.

rayColourOuter (colour) **-** sets the colour at the end of the rays.

Shards

+ shard - open this to display the shard parameters.

shardForm (int) **-** controls whether to render a shard or not.

0 = none
1 = shard

shardOffset (float) **-** shifts the position of the light shard along the line between the rays centre and pivot positions.

shardRotation (float) - controls the rotation of the shard about its centre.

shardSize (float) - controls the length of the shard.

shardCoreSize (float) - controls the size of the central bulge.

shardWidth (float) - controls the thickness of the shard.

shardGain (float) - controls the brightness of the light shard.

shardBlend (int) - controls how to blend this component with the others.

shardColourCore () - sets the colour of the glow.

shardColourOuter () - sets the colour of the shard.

Polygons

+ polys - open this to display the polygon parameters.

polysForm (int) - sets the type of polygon rendered.
0 = none - switches off the rendering of polys.
1 = chromatic - renders rainbow coloured polys.
2 = normal - renders two colour polys.

polysCount (float) - sets the number of polygonal shapes rendered between the centre and pivot.

polysOffset (float) - shifts the position of the polygons along the line between the centre and pivot.

polysSize (float) - controls the size of the shapes. Increase this parameter for bigger circles and polygons.

polysCoreSize (float) - sets the radius of the circle drawn at the centre of the polygon.

polysSizeVariance (float) - controls how much the size of the shapes varies from the value set by polysSize.

polyRotation (float) - controls the rotation of the polygons.

polysSides (float) - sets the number of sides of the polygons.

polysCurvature (float) - controls the bending of the lines drawn between the polygon vertices. Increase this value to bow out the poly sides. High values will render circles.

polysSpacing (float) - controls the distance between each shape. Increase this parameter for more widely spaced circles and polygons.

polysSoftness (float) - controls the edge softness of the polygons.

polysSeed (float) - generates a random number sequence used to position the polygons.

polysGain (float) - controls the brightness of the polygons.

polysGainVariance (float) - controls the deviation in polygon brightness. A value of zero will ensure that all polygons are the same brightness.

polysBlend (float) - set how to composite the polygons with the other lens flare components.

polysColourVariance (float) - controls the deviation in the colour of the polygons from the polysColourCore and polysColourOuter parameters.

polysColourCore (colour) - inner colour.

polysColourOuter (colour) - outer colour.

Rings

+ rings - open this to display the ring parameters.

ringForm (float) - sets the type of ring drawn. See Figure 27 on page 18.

0 = **none** - switches off the ring.
1 = **ring** -
2 = **chromaticRing** - renders a rainbow coloured ring intersected with dark rays drawn from the centre of the ring.
3 = **chromaticRadial** - renders a rainbow coloured ring intersected with dark rays drawn from the ray centre.
4 = **chromaticRipple** - renders a rainbow coloured ring intersected with evenly spaced dark rays drawn from the centre of the ring.
5 = **arc** - renders a two colour arc.

Note *The difference between chromaticRing and chromaticRadial only becomes apparant when the ringOffset is non zero.*

ringOffset (float) - controls the shift in position of the ring from the ray centre to pivot position.

ringSize (float) - sets the radius of the ring.

ringWidth (float) - sets the thickness of the ring.

ringRotation (float) - controls the rotation of the ring.

ringSpokeCount (float) - essentially controls the amount of detail in the chromatic rings. If the detail is set to 1, a smooth rainbow coloured ring is drawn. As the detail is increased soft gaps appear in the ring. As the detail increases further the gaps become more numerous and are thinner.

Note *This parameter has no effect if the ringForm is set to ring or arc.*

ringGain (float) - sets the brightness of the ring.

ringBlend (float) - sets how to composite the ring with the other lens flare components.

ringColourCore (float) - inner ring colour.

ringColourOuter (float) - outer ring colour.

+ blending (group) - sets how to mix between the image effect and its original source. (See “Blending” on page 3.)

Hints & Tips

The introduction of a lens flare when creating a scene with a bright light source is quite common. It can play an important part in making the scene look right.

Real lens flares can be shot against a black background and composited into your scene. However, it can be tricky and expensive to match the camera moves of the two sequences so that the lens flare appears correct. Digital lens flares are fast and can easily be animated to track the movement of the light source in your scene.

Lens flares occur when a bright light is shone directly into the camera lens. Each lens flare has a bright highlight caused by the overexposure of the light on the film and a trail of polygons caused by the multiple reflections of the light rays in the lenses that form the focussing assembly of a camera. The shape of the iris that forms the camera’s aperture is responsible for the shape of the polygons formed in the lens flare. Since lens flares are constructed inside the camera, when you come to digitally creating them they should always be composited over everything else in your scene.

The precise form of the lens flare comes from the lens properties and not the light source. This is particularly apparent for anamorphic lenses which produce horizontal lens flares. These can be recreated using the Light Shard element of the LensFlare.

You should also be aware that the polygons will move at different speeds relative to each other whenever the light source or camera is moving. This is caused by the different position of the lenses within the lens assembly. When animating T_LensFlare you should keep the pivot position static in the centre of the image and animate the highlight position. The relative positions of the rings and polygons will automatically animate in relation to these two coordinates.

It is worth spending time looking at real lens flares to get a feel for the shapes, colours and movement. Just spending an evening in watching television will doubtless prove fruitful in this quest. You should note that lens flares have very subtle imperfections and tend to flicker over time.

You can use T_LensFlare to create rainbows. switch eveything off apart from one chromatic ring.

Figure 28. Simulated rainbow using T_LensFlare.

T_MatteTool

Description

T_MatteTool provides a suite of tools to manipulate mattes, in particular, the growing and eroding of edges with *subpixel* precision. However, it excels at providing a comprehensive set of tools for manipulating mattes pulled from blue and green screen keyers. Tools include matte clean-up while preserving edge detail.

Figure 29. Matte Input.

Figure 30. Grow.

Figure 31. Shrink.

Figure 32. Halo.

Inputs

T_MatteTool takes the following input.

- source (required)

Parameters

components - sets the channel(s) to process.

mode - controls how to manipulate the matte.

- **shrink/grow -** positive values grow the matte edges. Negative values erode the matte edges.
- **halo In/Out -** creates a line on the inside (or Outside) edge of the matte.

- **halo -** creates a line centered on the edge of the matte.

Figure 33. Split Screen showing Halo Out in orange and Halo In in white.

shape - sets the profile of the filter used to erode/grow the matte edges.

- **circle -** corners are rounded off with this algorithm.
- **square -** sharp corners are preserved with this algorithm.

xPixels - controls the amount of horizontal eroding or growing of the matte edges. Negative values erode the matte. Positive values grow the matte.

yPixels - controls the amount of vertical eroding or growing of the matte edges. Negative values erode the matte. Positive values grow the matte.

xSoftness - controls the amount of horizontal blurring applied to the matte.

ySoftness - controls the amount of horizontal blurring applied to the matte.

despeckleBlack (float) **-** increase this to remove black pixels in white areas while preserving edges. This works by growing the surrounding white pixels into the black spots.

despeckleWhite (float) **-** increase this to remove white pixels in black areas while preserving edges. This works by growing the surrounding black pixels into the white spots.

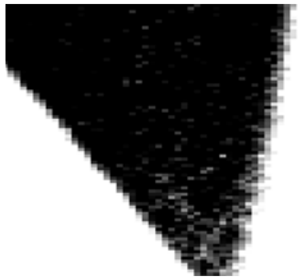
Figure 34. despeckleWhite = 0.


Figure 35. despeckleWhite = 1.

ClipMin **-** pixels at or below this luminance value are set to black. When compositing, this parameter can be used to improve the background image if parts of the foreground are showing through.

ClipMax **-** pixels at or above this luminance value are set to white.. When compositing, this parameter can be used to firm up the centre of the matte making it less transparent to the background. Increasing this value too much will affect the edges of your matte. Clip rollback should be used to compensate.

clipRollback - controls the amount of erosion of the edges of the black threshold matte when Clip Min is used to remove dust in the background.

negate - switch this on to invert the matte.

halosCutBlurs - this cuts into the inside or outside edges of softened In/Out halos.


Figure 36. Unprocessed matte.


Figure 37. Halo In/Out, Pixels = 6.

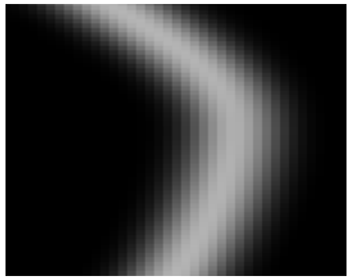
Figure 38. Softness = 7.

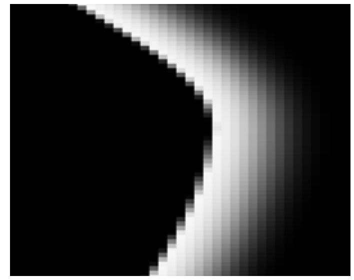
Figure 39. halosCutBlurs.

+ sourceEdges (group) - this plug-in may need access to pixels which lie off the edge of the image. The parameters in this group control which pixels are used in these calculations. (See “Source Edges” on page 5.)

T_Median

Description

T_Median is a subpixel accurate N x N median filter. It is useful for cleaning up isolated dots and noise.

Inputs

T_Median takes the following input.

- source (required)

Parameters

channels (float) - sets which channels to process.

xPixels (float) - horizontal size of the filter.

yPixels (float) - vertical size of the filter.

+ sourceEdges (group) - this plug-in may need access to pixels which lie off the edge of the image. The parameters in this group control which pixels are used in these calculations. (See “Source Edges” on page 5.)

T_NightSky

Description

T_NightSky renders the brightest 9000 stars visible from Earth. It uses real data so you get to see the constellations in their correct positions. There are controls to manipulate the camera and twinkle

Figure 40. T_NightSky showing The Plough

the stars.

Inputs

T_NightSky takes the following input.

- background (optional)

Parameters

outputDepth (int) - sets output colour depth.
0 = Auto
1 = 8 bit
2 = 16 bit
3 = float

nightSkyWidth (int) - sets the width of the output image.

nightSkyHeight (int) - sets the height of the output image.

aspect (float) - controls the horizontal and vertical weighting of the effect.

gotoConstellation (string) - sets the camera parameters to display some well-known constellations.

- orion
- southernCross

- thePlough
- cassiopia
- pegasus
- cygnus
- auriga
- taurus
- alphaCentauri
- scorpio
- hercules
- canisMajor

altitude (float) - controls the elevation of your telescope.

azimuth (float) - controls the horizontal direction of your telescope.

twist (float) - controls the rotation of your telescope about an axis along the length of the telescope.

fieldOfView (float) - controls how much of the night sky you can see. Increase this for a wide angle lens.

Stars

+ **stars** - open this to display the star parameters.

starShape (string) - sets the shape of the stars.

maxSize (float) - controls the maximum size of a star.

sizeSpread (float) - controls the range of star sizes in the image. A value of zero will force every star to be the same size. Increasing this value will increase the difference in scale between the largest and smallest stars.

brightness (float) - controls the overall brightness of the stars. The brightness of individual stars may vary.

luminanceSpread (float) - controls the range of star brightnesses in the image.

saturation (float) - controls the colour saturation of the stars. A value of zero will render stars with no colour.

flickerRate (float) - controls the speed of the luminance variations of the stars.

flickerAmount (float) - controls the amount of luminance variation during flickering.

jitter (float) - controls the small random variations in position of the stars.

+ blending (group) - sets how to mix between the image effect and its original source. (See “Blending” on page 3.)

T_Rays

Description

T_Rays creates a backlit ray effect. The rays are sourced from selected areas of the image's luminance, or from a matte. The colour of the rays can come from the source image or a fixed colour.

Figure 41. T_Rays through a stained glass window

Note *Before altering the ray parameters you should view and manipulate the matte as this defines the source of any rays generated. White areas of the matte will emit rays, black areas will not.*

Inputs

T_Rays takes the following inputs.

- source (required)
- matte (required)

Parameters

xCenter (float) - sets the horizontal position of the source of the rays.

yCenter (float) - sets the vertical position of the source of the rays.

factor (float) - controls the strength of the rays. Increase this for longer rays.

falloff () - controls how the rays fade as they get longer.

- **linear** - equal fade over the length of the rays.
- **exponential** - rays fade more smoothly at their ends.

colouring (string) - controls whether to take the colour of the rays from a fixed colour or from the colours in the image.

- **source -** the colours of the rays are taken from the source image.
- **colours -** the colours of the rays are taken from the T_Rays.colourN parameters.

nColours (int) - sets the number of colours in the rays.

Note *This is only active if colouring is set to colours.*

T_Rays.colourN (colour) - sets the colours in the rays if colouring is set to colour. Up to 5 colours can be used.

matteChannel (int) - controls which colour channel is used to source the rays.

- **grey -** luminance is used.
- **alpha -** the alpha channel is used.

seed () - sets the randomness of the rays.

filtering (group) - sets the quality of the filter used when processing the effect. (See “Filtering” on page 2.)

Scintillation

+ **scintillation** (group) - this group cuts the smooth rays.

scintillates (boolean) - switches scintillation on and off.

scintillationAmount (float) **-** sets the contrast of the scintillation lines. Low values give a smoother result.

scintillationSpeed (float) - sets the rate of movement.

scintillationDetail (float) **-** sets the complexity (number of lines) of the scintillation.

+ blending (group) - sets how to mix between the image effect and its original source. (See “Blending” on page 3.)

T_Starburst

Description

T_Starburst adds sparkle rays to highlights. The optional attenuation


Figure 42. Car headlamps

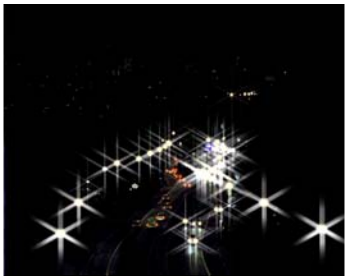
Figure 43. T_Starburst

matte is used to affect the gain on the sparkles.

Inputs

T_Starburst takes the following inputs.

- source (required)
- matte (optional)

Parameters

presets (string) - popular looks to get you started.

numberOfSpokes (float) **-** sets the number of rays on each sparkle.

xPixels (float) - sets the length of the rays.

yPixels (float) - sets the length of the rays.

xGap (float) **-** controls the distance between the source of the rays and the rays themselves.

yGap (float) **-** controls the distance between the source of the rays and the rays themselves.


Figure 44. Gap = 0.

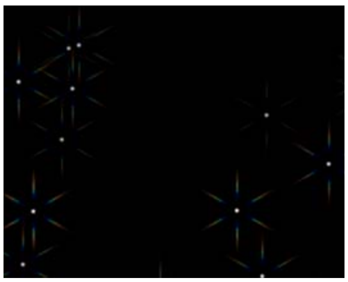
Figure 45. Gap = 20.

gain (float) - controls the brightness of the starburst.

threshold (float) - sets the luminance level above which sparkles are added.

aberration (float) - colours the rays.

keepSpokeBrightness (boolean) - switch this on the preserve the ray brightness as the length of the rays increases.

rotation (float) - controls the rotation of each sparkle.

maskWith (string) - sets how to use the matte input to alter the gain on the starburst.

- none
- luminance
- alpha
- negativeLuminance
- negativeAlpha

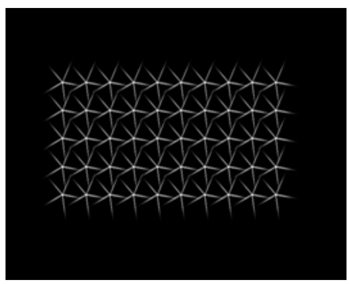

Figure 46. Dots with Starburst

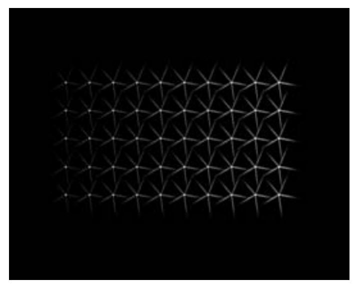

Figure 47. Starburst using a ramp matte to attenuate the gain

+ blending (group) - sets how to mix between the image effect and its original source. (See “Blending” on page 3.)

+ sourceEdges (group) - this plug-in may need access to pixels which lie off the edge of the image. The parameters in this group control which pixels are used in these calculations. (See “Source Edges” on page 5.)

Appendix A

GENARTS TINDER PLUG-INS

END USER SOFTWARE LICENSE AGREEMENT

By installing GenArts Tinder plug-ins, you agree on behalf of all the users of this installation, to the terms and conditions of this agreement. If you do not agree, please do not install this software.

You are hereby granted a non-exclusive, non-transferable license to use this software on a temporary trial basis. If purchased, you may continue using it under these same conditions. For each license purchased, you may use the software on a single computer. If you install the software and license more than once, you may only use it on one computer at a time. If you uninstall the software and properly register that uninstall, you may then reinstall the license on a new computer. You may also use the software for network rendering on additional computers if your host application supports it, and the plug-ins software supports it with that host application, and you initiate the network rendering from a licensed computer.

This software and documentation is owned by GenArts, Inc. and is protected under copyright law. However, you may copy it onto and amongst computers at your facility containing Shake or compatible products.

You may not reverse engineer, decompile, or disassemble this software.

You should test this software thoroughly before depending on it, since like any software it may contain bugs. Although efforts will be made to document and/or repair problems as they are found, this software and documentation are provided "as is" without warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the results and performance of the software is with you. In the event of trouble, your exclusive remedy shall be repair or replacement of this software. GenArts, Inc. will not be liable for any consequential, incidental, indirect or special damages arising from the use of this software or documentation under any theory of liability, whether contract, strict liability or tort including negligence or otherwise. In any event, GenArts' total liability shall not exceed the amount paid by you for this software.

Index