

Tinsel user guide

Tinsel is an **LED-outline effect for Resolume Arena and Avenue**, as an FFGL plugin. It finds the outlines in whatever is on the layer and hangs a string of virtual lamps along them, then runs the kind of pattern an LED controller runs: chases, comets, twinkles, fire, a scanner, a wipe. Point it at a logo and it becomes a neon sign, a theatre marquee, or a Christmas tree.

The idea it is built on is the one an LED controller is built on: **a pattern never chooses a colour**. It decides *where in the palette* each lamp sits and *how bright* it is, and the palette turns the first of those into RGB. That is why twenty patterns and sixteen palettes are worth more than thirty-six of either — Comet on **Christmas** is a red-and-green chaser, Comet on **Frost** is a shooting star, and neither is the other with a different colour on it.

Before you rely on this: the pattern library is verified numerically. It exists twice, once in C++ and once in GLSL, and an offline harness runs the shader at one pixel per lamp and compares the result against the C++ — **527,100 comparisons with zero disagreements**. All 31 controls are separately confirmed to change the picture. Both the macOS universal bundle and the Windows x64 DLL build in CI.

Still open: it has **never been loaded into Resolume**. Everything about how the controls *present* in the host is untested — whether the Colour 1 and Colour 2 triples show as swatches, and whether five groups of 31 controls reads sensibly in the inspector. Performance has never been measured. Try it on a spare layer before you put it in a show.

This codebase was created with AI assistance, directed and reviewed by a human author.

Installing

Drop the plugin into Resolume's FFGL folder and restart Resolume:

```
macOS    ~/Documents/Resolume Arena/Extra Effects/  
         (or /Users/Shared/Resolume Arena/Extra Effects/)  
Windows  %USERPROFILE%\Documents\Resolume Arena\Extra Effects\
```

Avenue uses the same layout under its own folder name. Tinsel then appears in the effects browser.

Needs Resolume Arena or Avenue 7.3.1 or newer.

On macOS the build is unsigned, so Gatekeeper may quarantine it:

```
xattr -dr com.apple.quarantine ~/Documents/Resolume\ Arena/Extra\ Effects/Tinsel.bundle
```

OpenFX hosts (Resolve, Vegas, Nuke, Natron)

Tinsel also ships as an OpenFX plugin — same patterns, same palettes, same controls. Copy `Tinsel.ofx.bundle` from the `-ofx-` download into the OpenFX folder and restart the host:

```
macOS    /Library/OpenFX/Plugins/  
Windows  C:\Program Files\Common Files\OpenFX\Plugins\
```

One difference worth knowing: the Stability filter is rebuilt from a short window of previous frames each render, so very high Stability settings cost render time and are truncated slightly compared with the Resolume build.

Start here: get the outline right first

Everything downstream is lighting an outline, so **set the outline before you touch anything else** — and set it while you can see it.

Put **Background** on **Edges**. That shows the detected mask in plain white, with no lamps, no colour and no glow in the way. Judging a threshold through a layer of twinkling bulbs is guesswork.

Then:

1. **Detect On** — what "different" means between two pixels. This matters more than any of the other edge controls.
 - **Luma** — ordinary brightness edges. The right default for footage.
 - **Alpha** — for a logo delivered with transparency. Its alpha channel already contains a perfect outline, and running a brightness detector over it instead throws away the only clean signal in the frame.
 - **Chroma** — the boundary between two colours of *equal brightness*. A luma detector is completely blind to these, and brand artwork is full of them.
 - **Luma or Alpha** — both at once, for artwork that could arrive either way. Slightly less contrast on an opaque clip than plain Luma, so you may want a little more Sensitivity.
2. **Sensitivity** — how strong a boundary has to be to count. A clean black-to-white step measures 1.0, so the whole useful range for real footage sits low.
3. **Detail** — what *scale* the detector works at. Low finds every pixel of sensor noise; high finds the shape of a logo and ignores the texture inside it. On any real footage this is the control that separates "outline" from "confetti".
4. **Thickness** — widens the mask. A thin outline gives small tidy lamps; a thick one gives fat ones with more room for the glow to sit on.

5. **Softness** — how abrupt the threshold is. Wound right down, Sensitivity becomes a control that does nothing and then everything, and every diagonal aliases.

Then put **Background** back where you want it.

Stability, for video

Stability is a filter over time, and it is the control that makes this usable on footage rather than only on stills.

What goes wrong on video is not that edges *move* — it is that they **drop out**. A boundary grades through the threshold for a single frame, or sensor noise sits on a nearly-flat gradient, and the lamp blinks. So the filter is deliberately lopsided: it rises almost instantly, and falls slowly. An edge that appears is believed at once, so nothing is dragged behind a moving subject; an edge that vanishes is given a few frames to come back.

Wind it up until the flicker stops and no further. At the top of its range a lamp hangs on for about a second after its edge has gone, which starts to read as a ghost.

On a still image it does nothing at all, and that is correct — there is nothing to filter.

Where the string runs

Layout decides the path the virtual strip takes across the frame.

- **Spiral** (*default*) — goes round and climbs, which is what a string of lights actually does when someone puts it on a tree. **Turns** sets how tightly.
- **Angle** — once round the artwork is one strip. A chase runs round the outline like a clock hand. This is the marquee look, and it is usually the cleanest on a logo.
- **Linear** — a straight projection across the frame, so a chase becomes a wipe. **Direction** sets the angle.
- **Radial** — rings running outward from the middle.
- **Random** — every lamp stays where it is, but its place in the running order is shuffled. This is a strip wired in an arbitrary sequence, not lamps scattered at random.

Turns is worth understanding, because it is the difference between lamps and streaks. A lamp is a band laid across the strip's path. Where that band lies *along* the outline instead of across it, the lamp gets drawn out into a smear. A tightly wound spiral meets a round logo at a shallow angle nearly everywhere, so it smears; wound loosely it crosses close to square, and lamps read as lamps. If your lamps look like scratches, turn **Turns** down.

Lamps is how many, from 8 to 1000. **Lamp Size** is each one's radius, in pixels — so a lamp is the same size wherever it lands in the frame.

Lamp Size is also how you get a rope. Wind it up past half the lamp spacing and the lamps run into one another and the strip becomes a continuous glowing line. That is the neon-sign look, and it is the top of this one control rather than a separate mode. Wind it down for discrete bulbs with dark gaps between them.

Reverse flips which way round the strip is wired. Use this rather than trying to run Speed backwards: for a comet the two are not the same thing, because reversing time leaves the tail on the same side and the comet appears to travel tail-first.

The patterns

Speed is in cycles per second, and **zero means frozen** — useful for a static string of lamps.

Intensity is each pattern's second knob and means whatever that pattern needs it to.

Pattern	What it does	What Intensity changes
Solid	Every lamp on, one palette colour	—
Gradient	The palette laid along the strip, still	—
Rainbow	The palette laid along the strip, moving	—
Colour Loop	The whole strip one colour, walking the palette	—
Chase	Groups of lit lamps running along	How many chasers, and how tight
Theater Chase	Every n th lamp lit, stepping one at a time	The spacing, 2 to 6
Running Lights	A smooth sine wave travelling along	How many waves
Comet	A bright head with a fading tail	Tail length
Meteor	A comet whose tail breaks up into debris	Tail length
Larson Scanner	A head that bounces end to end	Tail length
Colour Wipe	Fills in one colour, then back in the other	—
Twinkle	Each lamp fades up and down at its own rate	How many are lit at once
Sparkle	Brief random flashes on darkness	How many
Glitter	Brief random flashes over a lit strip	How many
Fire Flicker	Per-lamp flicker, hotter lamps further up the palette	Flicker depth
Breathe	The whole strip swelling and fading	—
Strobe	Everything on, briefly, once per cycle	Flash length
Dissolve	Lamps swap between two colours in random order	—
Colorwaves	Colour and brightness undulating out of step	—
Fairy Lights	Sparse, slow, warm. The classic fairy light	How many lamps take part at all

Colour Loop reads as nothing at all on the Mono palette, which is correct rather than broken — it walks the palette, and that palette is one colour.

Colour

Palette is where a lamp's colour comes from.

The first two are drawn from the colour controls: **Colour 1** on its own, and **Colour 1 > 2** as a blend between the two. The other fourteen are fixed gradients — Rainbow, Party, Christmas, Candy Cane, Warm White, Frost, Fire, Ocean, Forest, Sunset, Cyberpunk, Gold, Magenta and Mono.

Spread is how many times the palette is laid across the strip. It only ever affects colour, never where anything is, so turning it up recolours a chase without moving the chase.

Saturation pulls towards white rather than towards grey, so at zero you get white bulbs. **Brightness** goes past 1.0 on purpose: a bulb that is not clipping does not look like a bulb.

Source Tint multiplies each lamp by the artwork underneath it, so a logo can light its own outline in its own colours.

Output

Glow is the halo, and it is *added* rather than blended — a bulb's halo is light arriving on top of whatever is already there. **Glow Size** sets how far it spreads.

Background decides what the lamps sit on:

- **Black** — lamps only, on solid black.
- **Source** — lamps added over the original clip.
- **Dimmed Source** — the clip knocked back by **Dim**, with the lamps over it. Usually the most useful of the three on real content.
- **Transparent** — lamps only, with everything else transparent, so the layer below shows through.
- **Edges** — the detected mask in white. Not a look; it is how you set the edge controls.

Mix fades the whole effect back towards the untouched clip.

If it looks wrong

Nothing happens at all. Check **Background** → **Edges** first. If the mask is black, the detector is finding nothing: lower **Sensitivity**, or change **Detect On** — a logo on transparency needs **Alpha**, and two colours of the same brightness need **Chroma**.

The whole frame lights up. Sensitivity is too low, so the detector is finding noise. Raise it, and raise **Detail** so it works at a coarser scale.

The lamps are scratches, not dots. Turn **Turns** down, or switch **Layout** to **Angle**. See the note under "Where the string runs".

Everything flickers on video. Raise **Stability**. Also raise **Detail** — most edge flicker is the detector finding texture rather than shape.

It looks like a glowing line, not lamps. **Lamp Size** is past half the lamp spacing. Turn it down, or raise **Lamps**.

A chase doesn't follow my logo's outline. It cannot, and this is the honest limit of this version. The strip is a *field* across the frame, not a traced contour, so a chase climbs the shape the way a string wound round a tree climbs it rather than following a letterform round its own curve. Separate parts of a shape are not wired in series either. Contour tracing is planned.

Diagnostics

If the effect loads but does nothing at all, the log will say which shader failed to compile:

```
macOS    ~/Library/Logs/tinsel/tinsel.YYYY-MM-DD.log
Windows  %LOCALAPPDATA%\tinsel\logs\tinsel.YYYY-MM-DD.log
```

It records the GL vendor, renderer and version alongside, because a shader that compiles on one machine and not another is a driver answer rather than a source one.

Tinsel · v0.1.0 · guide updated 2026-08-03 · stoatworks-labs.com/software/tinsel/

Latest version of this guide: stoatworks-labs.com/software/tinsel/guide/ · Source and issues:

<https://github.com/stoatworks-labs/tinsel>