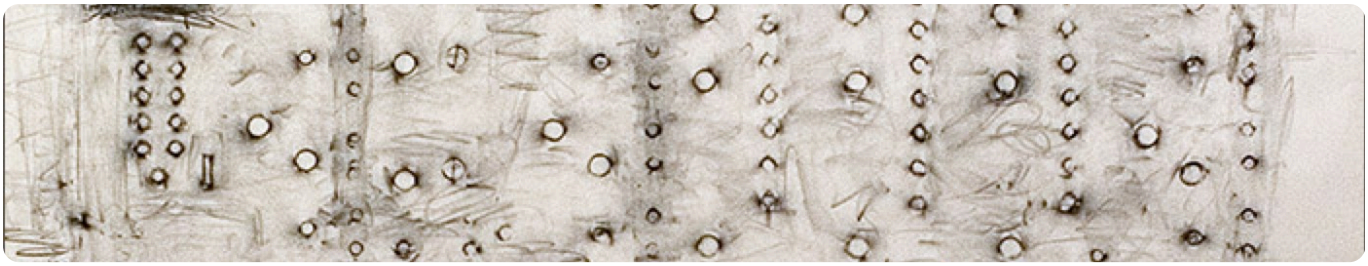
SANDIN FX

Real-time video effects synthesizer for macOS — By Mescalina

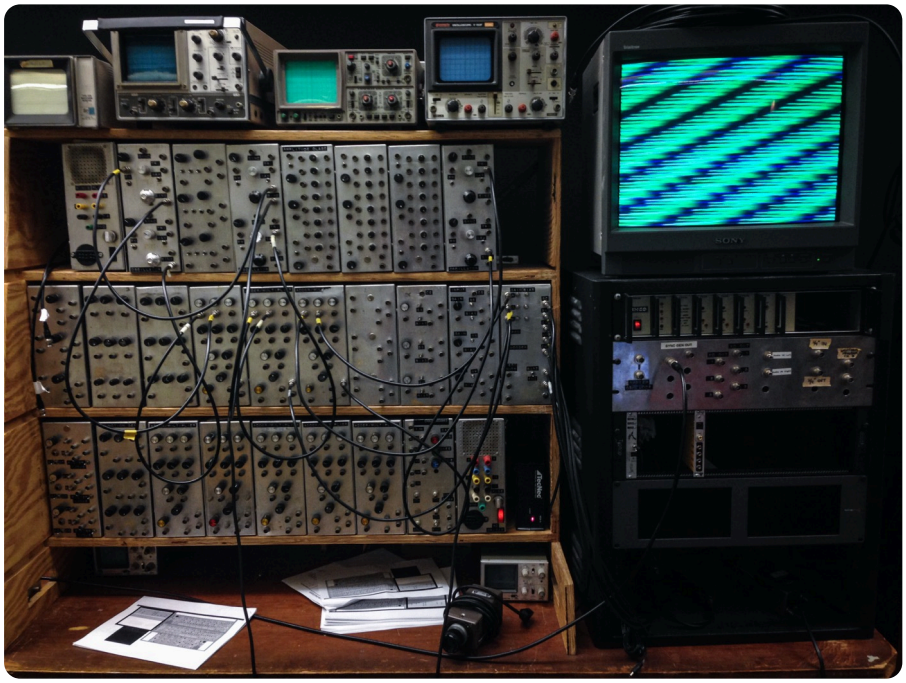
An instrument, not an editing program: you turn knobs and the image transforms, live, like on the analog video synthesizers of the 1970s that inspired it.



1. The history — the Sandin Image Processor

Between 1971 and 1973, **Daniel J. Sandin** built in Chicago a machine that would mark the history of video art: the **Image Processor**, often described as the video equivalent of a Moog synthesizer. The idea: apply the principles of sound synthesis to the image — rather than playing with sound waves, you play with video signals, in real time.

Its strength: a **modular, analog architecture**. Not a single machine but independent aluminium boxes — knobs, video jacks, patch cables — assembled à la carte: input modules (camera, video recorder); processing modules — *Adder/Multiplier* to mix two signals or push the contrast, *Comparator* for luminance-threshold keying, *Differentiator* to sharpen edges, *Function Generator* for electronically generated patterns — and output modules, such as the *Color Encoder* which, in the mid-70s, turned a black-and-white signal into a colour image. Sandin even distributed the plans freely: the spirit of free software ahead of its time.



The Sandin Image Processor: the aluminium modules — oscillators, Adder/Multiplier, encoder — patched with cables in their wooden racks, the monitoring oscilloscopes, and the monitor where the image is born. Every module of Sandin FX descends from one of these drawers.

2. GRASS, DeFanti and the Electronic Visualization Laboratory

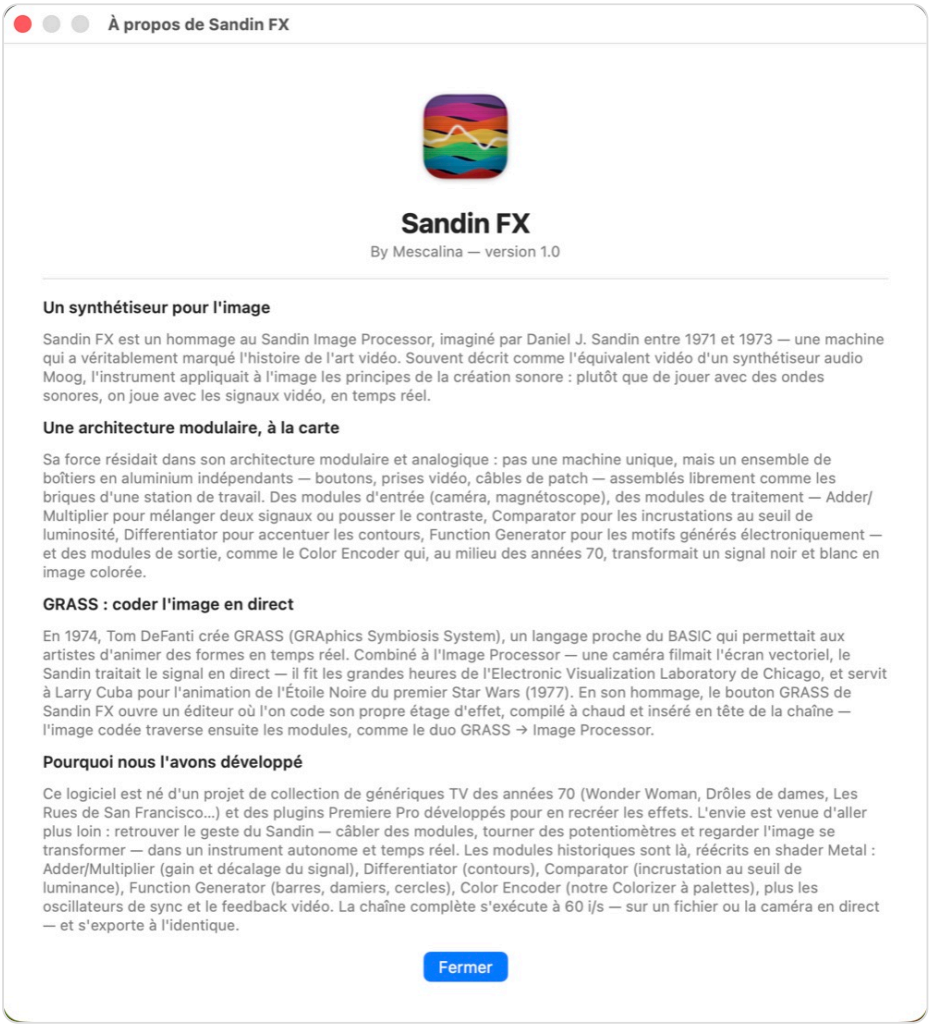
In 1974, **Tom DeFanti** created for his thesis the **GRASS** language — *GRAphics Symbiosis System* — close to BASIC, designed so that artists without technical training could animate 2D shapes in real time: rotation, scale, translation, colour. The combination made it famous: GRASS generated the image on a Vector General vector display, a camera filmed the screen, and the Sandin Image Processor processed the signal live. Installed at the Circle Graphics Habitat of the University of Illinois at Chicago — the future **Electronic Visualization Laboratory (EVL)** — the system enabled the live audiovisual performances of the *Electronic Visualization Events*, mixing graphics, image processing and sound synthesis.

Popular consecration in 1977: it was with GRASS that **Larry Cuba** made the vector animation of the Death Star — *“Attacking the Death Star will not be easy”* — in the first *Star Wars*. The language then evolved (GRASS3 by Nola Donato, Zgrass on the chip of the Bally Astrocade console, and the Datamax UV-1, one of the first dedicated graphics workstations).

3. The genesis of Sandin FX

This software was born of a detour. At the start: a collection of 1970s TV title sequences — *Wonder Woman*, *Charlie's Angels*, *The Streets of San Francisco*, *Banacek* — and the trailer of *Les Baronnes*, whose effects were dissected frame by frame to recreate their mechanics in four Adobe Premiere Pro plugins (duotone cast cards, diagonal-band reveal, starred title cards, pop silhouettes).

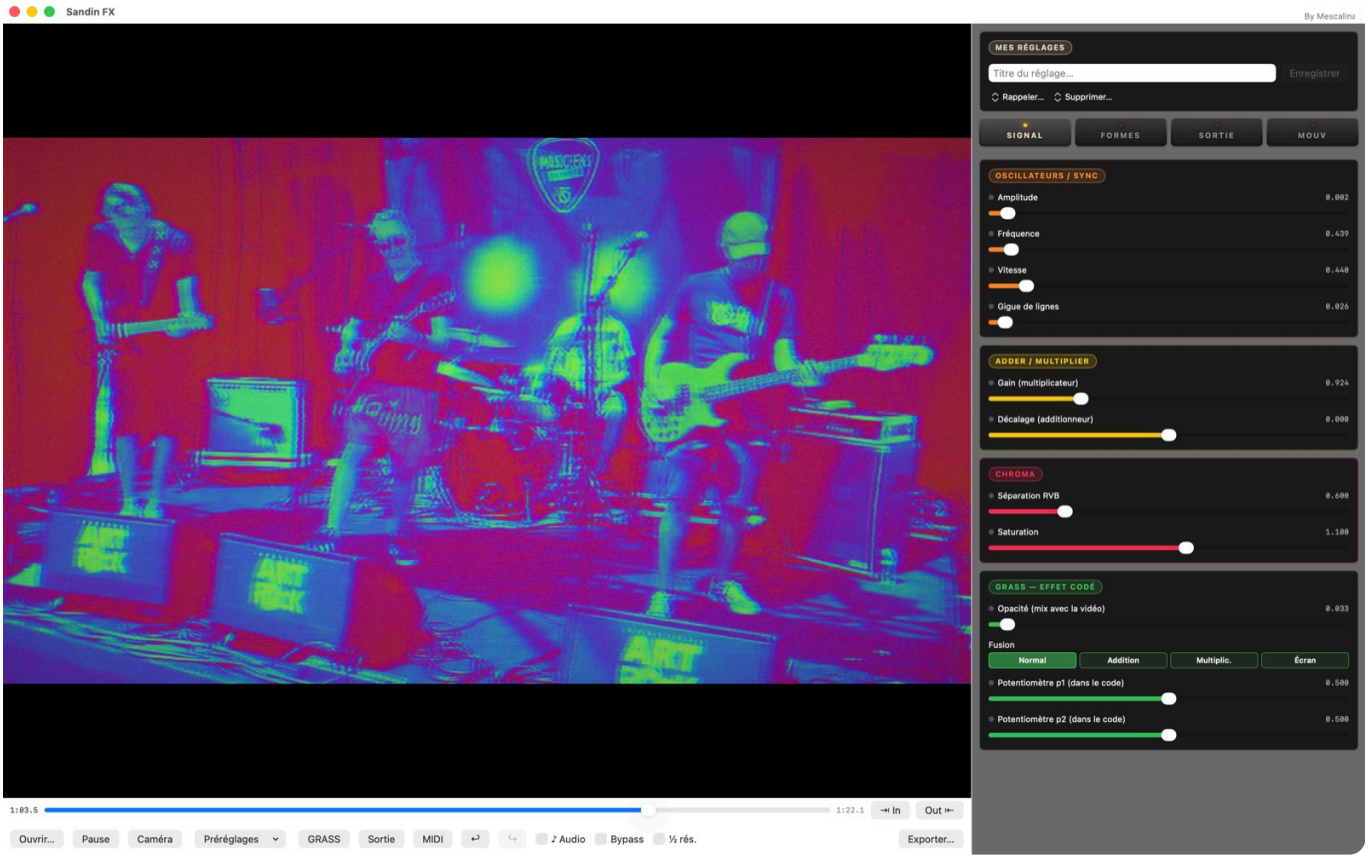
Then the wish to go further: to recover the **gesture** of the Sandin — patching modules, turning knobs, watching the image transform — in a standalone, real-time instrument, rather than dropping effects onto a timeline. Sandin FX is developed by **Mescalina** in Swift and Metal, native to macOS (a Linux edition, in C++ and OpenGL, shares the same effects chain and the same settings files).



The application's “About” window tells this lineage.

4. The software

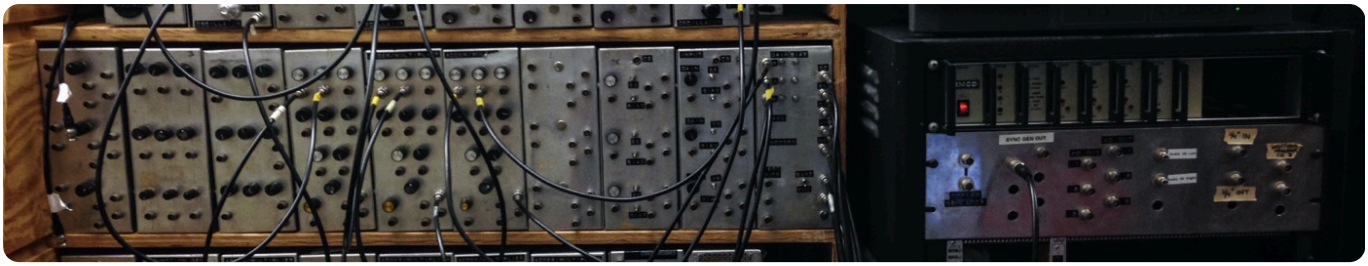
The whole effects chain fits in **a single Metal shader**, compiled at launch and executed pixel by pixel at 60 frames per second — on a video file (with timeline, In/Out points and loop playback) or on the **live camera**. Video feedback uses a ping-pong of textures: the output of each frame becomes the input of the next, hence the infinite trails. The export (H.264 or ProRes) replays the chain **identically**, driven by the timecode: what you see is what you render.



The interface: the real-time preview (here a concert processed in false colours), the timeline with In/Out points, the transport bar (camera, GRASS, output, MIDI, audio, bypass) — and the rack of coloured modules on the right, organised in four tabs with vintage buttons. Every slider carries its MIDI Learn dot.

Input signal (“Clean signal” preset)

The same, through the “Acid colorizer”



5. The modules — detailed catalogue and references

Seventeen modules, organised in three tabs that follow the path of the signal. For each one: what it does, and the historical object it refers to.

Signal tab — preparing and distorting the signal

Module	Reference	What it does
Oscillators / sync	The oscillators of the Sandin IP, and the “unlocked” picture of a detuned TV set	Horizontal undulation of the image driven by an oscillator (amplitude, frequency, speed) and random line jitter — the uncertain tracking of a tired video recorder.
Kaleidoscope	The mirror games and re-filming of 1970s video artists	Angular folding of the plane into N sectors (2-12) + rotation, applied *at the head of the chain*: everything downstream — shapes, colorizer, feedback — operates in the folded space, hence coherent mandalas that evolve with the feedback.
Rutt / Etra	**Rutt/Etra Scan Processor (1972)** — the other legendary video synth, which magnetically deflected the tube's scan (the aesthetics of Bowie's *Ashes to Ashes*)	Bright areas *lift the scan lines*: any video becomes a luminous wireframe relief (iterative resolution in the shader), with visible displaced lines and black out-of-frame.
Liquid warp	The analog “rubber sheet”: re-filming through distorting glass, bending the signal	Displacement of the image by an animated fluid noise — strength, scale and speed of the turbulence. The image ripples as if seen through water or heat.
Adder / Multiplier	The **Adder/Multiplier** module of the Sandin IP (mixing two signals, pushing the contrast)	Gain (multiplier) and offset (adder) of the signal — contrast and level, at the source.
Chroma	The chroma drift of analog consoles and the misalignment of the RGB guns in tubes	Separation of the red/green/blue channels (chromatic fringing on edges) and saturation.
GRASS	**Tom DeFanti's GRASS (1974)** — the language that generated the image sent into the IP	A *user-coded* effect stage (editor with syntax highlighting, hot compilation), inserted at the head of the chain, with opacity, four blend modes and two knobs p1/p2. Twelve examples included, dedicated PDF tutorial.

Shapes tab — analysing and generating

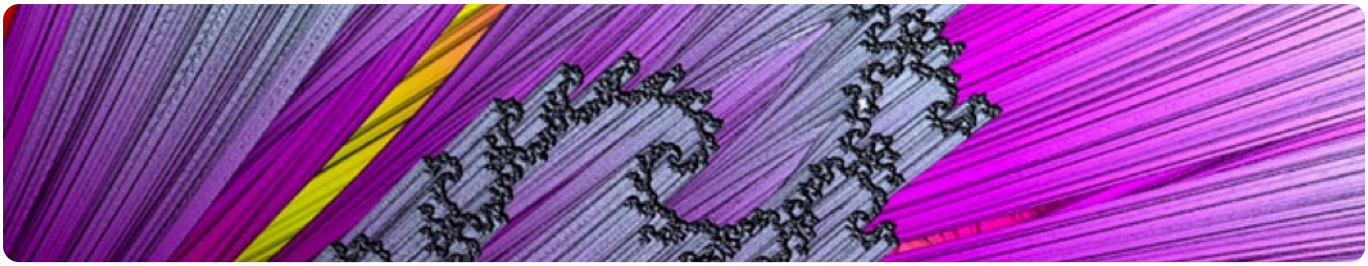
Module	Reference	What it does
Differentiator	The **Differentiator** module of the Sandin IP (derivative of the video signal)	Edge enhancement: the luminance gradient is fed back into the signal — the electronic “sharpen” filter.
Comparator	The **Comparator** module of the Sandin IP (threshold keying)	Binary keying at a luminance threshold, with adjustable transition softness — the image splits into clean zones, recolourable downstream.
Thresholds	Photographic solarisation (Man Ray) and the quantisation of the first colorizers	Solarisation (inversion above the threshold) and posterisation (2 to 10 levels) — the silkscreen look.
Function Generator	The **Function Generator** module of the Sandin IP, and the classic patch of two oscillators fed into each other	Electronic patterns (H/V bars, sinusoidal checkerboard, circles) that modulate the signal — plus a *second interference oscillator* at a nearby frequency: moirés and geometric beats.
Shapes — generator	GRASS animated 2D objects; the spiral trajectory quotes Sandin's *Spiral 5*	Random objects (circles, rings, squares, segments) that are born, grow and fade — density, size, speed, seed, blend. Random trajectory or *golden spiral* (born at the centre, escaping outwards). Deterministic: identical in the export.

Output tab — colouring, looping, texturing

Module	Reference	What it does
Colorizer	The **Color Encoder** of the Sandin IP and the Paik/Abe colorizers — colouring a black-and-white signal	Luminance drives a false-colour ramp. Seven palettes, three of them sampled from Sandin's works: *Spiral* (midnight blue → violet → pinkish white), *Julia* (magenta → yellow → silver) and *Quaternion* (black → violet → orange → cyan). Animated drift, switchable off.
Feedback	Camera-monitor video feedback — the queen technique of *Spiral 5*	Re-injection of the previous frame: zoom, rotation, *movable centre* (off-centre spirals), *rotating hue per echo* (each generation changes colour), adjustable persistence, four blend modes (Max, Add, Screen, Dissolve) and *RGB separation of the echoes* — uneven persistence of the “phosphors”, hence rainbow wakes.
Glow — halo	The halo of excited phosphors on a cathode-ray tube	Additive bloom of the bright areas (threshold, radius). Sampled from the previous frame: it covers everything — shapes, GRASS, feedback trails — at almost no cost.
Strobe — time freeze	The first frame freezes of frame buffers and the strobe of audiovisual performances (EVE)	The image only refreshes at the chosen rate (0.5-15 Hz); the rest of the time it replays the frozen frame — from a slight stutter to a near-photographic freeze, dosable.
Artifacts	The materiality of the signal: scan lines, tape hiss, vignetting and the phosphor mask of tubes	Scanlines, noise, vignette, and *CRT mask* — the red/green/blue triads visible up close on a cathode-ray screen (superb combined with the Glow).

Motion tab — playing time

The fourth tab does not process the image but **the way you play it**, echoing the voltage control of modular synthesizers: three **LFOs** (the Moog's low-frequency oscillators, applied to image parameters), **Gestures** (recording the knob performance, replayed and applied to the export), **audio reactivity** (sound drives the image — the spirit of the *Electronic Visualization Events* that mixed sound and visual synthesis) and **Morphing A → B** between two saved settings.

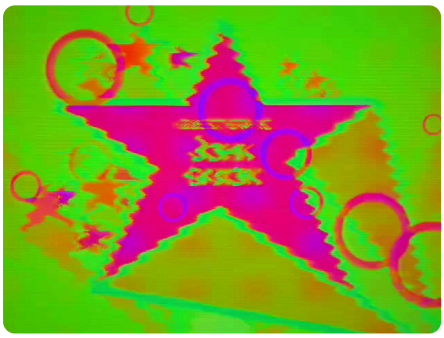


6. Gallery — eleven presets, one signal

Broken sync

VHS 1978

Solar psych



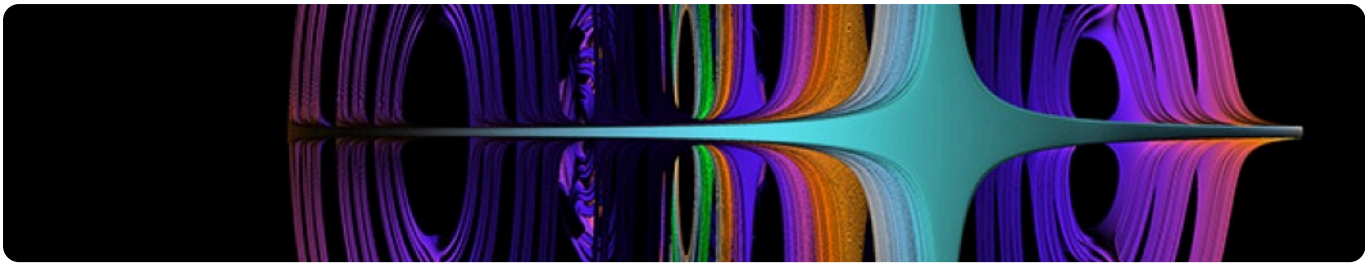
Total psychedelia (with shape generator)

Infrared nightmare

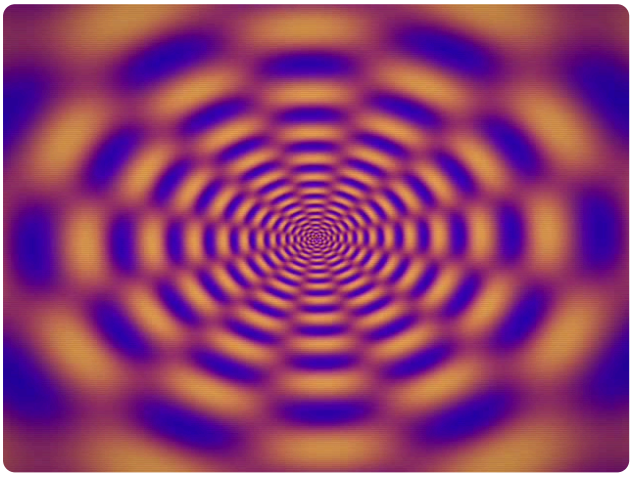
Liquid sheets (feedback)

7. A performance instrument

- **MIDI**: any USB controller, *MIDI Learn* pairing in two gestures, pickup mode — and **block navigation**: one knob selects the module (the tab follows, the block highlights), four knobs drive its parameters. Five knobs are enough for the whole program.
- **LFO**: three internal oscillators (sine, triangle, square, random) assignable to any parameter — the image moves on its own.
- **Audio reactivity** (behind an explicit switch): the microphone analysed as level + three bands, three sound → parameter routings. Shapes are born on the bass.
- **Gestures**: knob movements are recorded during playback, replayed in a loop, saved to files — and **applied to the export**: the performance becomes the render.
- **Morphing A → B**: parametric crossfade between two saved settings.
- **Output window**: the image alone, full screen on a second display or a projector, while the controls stay on the main screen. **A/B Bypass** to compare with the raw signal, **⌘Z** to recover.
- **Camera REC**: the processed output is recorded live, microphone included.



8. GRASS resurrected — coding the image



Pure generator coded in the GRASS editor (tunnel), processed by the feedback

In tribute to DeFanti's language, the **GRASS** button opens a code editor (syntax highlighting, line numbers) where you write your own effect stage in Metal — hot-compiled, inserted at the head of the chain, exactly like the GRASS → Image Processor duo of 1974. Twelve examples included, two knobs p1/p2 driveable by MIDI, adjustable opacity and blend, code embedded in the saved settings. A dedicated PDF tutorial teaches the language step by step (*GRASS-Tutorial-EN.pdf*).

9. Technical sheet

Platform	macOS 14+ (Apple Silicon and Intel with Metal) — Swift, SwiftUI, Metal, AVFoundation. Linux edition: C++17, SDL2, OpenGL 3.3, Dear ImGui, FFmpeg
Sources	Video files (timeline, In/Out, loop) and live camera
Rendering	Real time 60 fps, full or half preview resolution; export always at full resolution, deterministic
Export	.mov — H.264 (3 quality levels), ProRes 422, ProRes 4444; original audio preserved; In/Out range; cancellable
Command line	`SandinFX --render input.mp4 output.mov [preset]` — batch rendering without interface
User files	Settings, gestures, MIDI mappings and GRASS code as readable JSON/text in ~/Library/Application Support/SandinFX/
Privacy	Camera and microphone only on explicit user action; no network connection