Camlab User Manual

Table of Contents

1. Introduction

Camlab is a modular DSP pipeline builder designed for video experimentation, rapid prototyping, and real-time signal processing. It provides a flexible environment where developers can assemble complex processing chains from small, reusable DSP blocks, visualize data flow, and iterate quickly on new ideas. Whether you are exploring novel video effects, building research tools, or constructing high-performance real-time systems, Camlab offers a foundation that is both approachable and deeply optimized. A core strength of Camlab is its threaded, SSE3-optimized execution pipeline. Each DSP block is implemented with a focus on predictable performance: vectorized kernels accelerate per-pixel and per-sample operations, while the threaded scheduler distributes work across available CPU cores. This combination allows pipelines to scale gracefully with modern hardware, maintaining low latency even under heavy computational load. The result is a system that supports:

- High-throughput video processing — SSE3 vectorization enables wide-lane arithmetic, reducing the cost of filters, transforms, and color operations.
- Deterministic real-time behavior — threaded scheduling minimizes stalls and ensures that independent DSP blocks run concurrently without blocking the entire graph.
- Modular experimentation — developers can insert, remove, or reorder blocks without rewriting the underlying engine, making it ideal for prototyping new algorithms.
- Efficient resource utilization — the pipeline adapts to multi-core CPUs, balancing workloads to avoid bottlenecks and maximize frame-to-frame consistency.

This manual explains how to install, use, and extend Camlab. It also provides reference documentation for all available DSP blocks, along with guidance for building your own optimized modules. By combining a clean modular design with a high-performance execution core, Camlab enables both rapid experimentation and production-grade real-time processing.

2. Installation

System Requirements

- Windows 10 (version 1903 or later) or Windows 11 — 64-bit
- A compatible camera: ZWO ASI series, SVBony series, Windows webcam (WinMF), or a video file source
- Visual C++ Redistributable 2022 (x64)
- For H.264 / MJPEG streaming: an available network port (default H.264: 8080, MJPEG: 8081)

Installing from the Microsoft store

1. Open the Microsoft Store app on your Windows machine.
2. Search for "Camlab" in the store search bar.
3. Click on the Camlab app in the search results to view its store page.
4. Click the "Get" or "Install" button to download and install Camlab on your system.
5. Once installed, you can launch Camlab from the Start menu or by searching for it in the Windows search bar.

Camera Drivers

- **ZWO ASI cameras** — Install the driver for your camera available from the ZWO website.
- **SVBony cameras** — Install the driver for your camera available from the SVBony website.
- **Webcams / capture cards** — Use the standard driver supplied by the device manufacturer.
- **Video file source** — No additional driver required; supports common video formats via FFmpeg.

3. Usage

Application Pages

SD_Camlab is organised into several navigation pages accessible from the main window sidebar:

- **Camera** — Select and configure the camera source (device, resolution, exposure, gain). Supported sources: ZWO ASI, SVBony, Windows webcam/capture card, and video file playback.
- **Process** — Build and configure the DSP pipeline. Add, reorder, enable/disable, and configure individual DSP blocks.
- **View** — Preview the live processed output frame.
- **Settings** — Application-wide settings (performance, storage paths, etc.).
- **Log** — Runtime event and diagnostic log.
- **About** — Version information.

Basic Workflow

1. Open the **Camera** page and select your camera source from the dropdown. Set the desired resolution and frame rate.
2. Click **Connect** to begin streaming frames from the camera.
3. Open the **Process** page. The DSP pipeline is displayed as an ordered list of blocks.
4. Click **Add Block** and choose a DSP block from the menu. The block is appended to the end of the pipeline.
5. Use the block's parameter controls (sliders, toggles, buttons) to tune its behaviour in real time.
6. Enable or disable individual blocks using their on/off toggle without removing them from the pipeline.
7. Drag blocks to reorder them in the pipeline.
8. Open the **View** page to see the final processed output.

Pipeline Execution Model

Frames captured from the camera are passed sequentially through each enabled DSP block in the order they appear in the pipeline. Each block receives the output of the previous block as its input and produces a modified frame as its output. Blocks that are turned off are skipped entirely; the frame passes through unchanged. Processing is performed in a dedicated background thread to keep the UI responsive.

Saving Outputs

Use the **Save Photos**, **Snap Photo**, or **Save Movie** DSP blocks to capture the processed output to disk. Each block saves files into a dated sub-folder (`photos\` , `snapshots\` , or `movies\`) inside a user-selected root folder. Use the **Browse folder** control in each block to choose the root capture directory.

Network Streaming

Add the **Stream H264** or **Stream MJPEG** block to broadcast the processed output over HTTP. Once the block is enabled, open a browser on any device on the same network and navigate to `http://<host-ip>:<port>` to view the stream. Use the **Open browser** control to launch the default browser on the local machine automatically.

4. DSP Blocks

Each DSP block is documented below with its purpose, inputs, outputs, parameters, and an example pipeline snippet. All blocks receive and output an **RGB24** frame unless noted otherwise. Parameters are adjusted in real time via sliders or toggle controls on the Process page.

DSP Block Index

Bias Color

Description: Adds a signed constant offset to each RGB color channel independently. Useful for correcting color balance or shifting the black level per channel.

Inputs: RGB24 frame

Outputs: RGB24 frame with channel offsets applied (clamped to 0–255)

Parameters:

- **Bias_R** — Red channel offset. Range: −127 to 127. Default: 0.
- **Bias_G** — Green channel offset. Range: −127 to 127. Default: 0.
- **Bias_B** — Blue channel offset. Range: −127 to 127. Default: 0.

Example:

```
// Warm the image by lifting red and reducing blue
Bias Color → Bias_R: +30, Bias_G: 0, Bias_B: -20
```

Back to DSP Block index

Bias White

Description: Adds a signed constant offset equally to all three RGB channels. Brightens or darkens the image without changing its color balance.

Inputs: RGB24 frame

Outputs: RGB24 frame with uniform brightness offset applied

Parameters:

- **Bias** — Brightness offset applied to all channels. Range: −127 to 127. Default: 0.

Example:

```
// Lift a dark image by 40 counts
Bias White → Bias: +40
```

Back to DSP Block index

Blur

Description: Applies a Gaussian blur to soften the image and reduce high-frequency noise.

Inputs: RGB24 frame

Outputs: RGB24 blurred frame

Parameters:

- **Radius** — Blur kernel radius in pixels. Minimum: 1. Default: max_radius / 4.

Example:

```
// Pre-blur before delta detection to reduce noise
Blur → Radius: 3
Delta → Floor: 2
```

Back to DSP Block index

Crop

Description: Crops the frame to a sub-region defined by percentage-based coordinates. All subsequent blocks in the pipeline receive the smaller cropped resolution.

Inputs: RGB24 frame

Outputs: RGB24 cropped frame (smaller resolution)

Parameters:

- **X %** — Left edge of crop region as a percentage of frame width. Default: 0.
- **Y %** — Top edge of crop region as a percentage of frame height. Default: 0.
- **Width %** — Width of crop region as a percentage of frame width. Default: 100.
- **Height %** — Height of crop region as a percentage of frame height. Default: 100.

Example:

```
// Crop to the central 50% of the frame
Crop → X%: 25, Y%: 25, Width%: 50, Height%: 50
```

Back to DSP Block index

Delta

Description: Computes the absolute pixel-wise difference between the current frame and the previous frame. Static areas become black; moving areas are highlighted in proportion to their motion magnitude.

Inputs: RGB24 frame

Outputs: RGB24 frame showing inter-frame pixel differences

Parameters:

- **Floor** — Noise rejection threshold. Differences below this value are set to zero. Range: 0–64. Default: 1.

Example:

```
// Highlight motion in a static scene
Delta → Floor: 3
```

Back to DSP Block index

Delta Enhance

Description: Like Delta but with an additional gain multiplier applied to the difference signal, making faint motion more visible.

Inputs: RGB24 frame

Outputs: RGB24 amplified inter-frame difference frame

Parameters:

- **Floor** — Noise rejection threshold. Range: 0–64. Default: 1.
- **Gain** — Multiplier applied to the thresholded difference signal. Range: 1–64. Default: 1.

Example:

```
// Amplify subtle movement 8×
Delta Enhance → Floor: 2, Gain: 8
```

Back to DSP Block index

Discriminator

Description: Evaluates focus sharpness of each incoming frame using a variance-of-Laplacian metric and rejects frames that fall below a dynamic quality threshold. Blurry frames are replaced with the last accepted sharp frame, ensuring only high-quality frames continue down the pipeline.

Inputs: RGB24 frame

Outputs: RGB24 frame — either the current (sharp) frame or the last accepted frame if the current is blurry

Parameters:

- **Focus selectivity** — Percentage threshold above the observed minimum quality required to accept a frame. Higher values are more selective. Range: 0–90. Default: 10.
- **Audio beep for bad frames** — Play a beep when a frame is rejected as blurry. 0 = off, 1 = on. Default: 0.

Example:

```
// Accept only the sharpest 30% of frames
Discriminator → Focus selectivity: 70, Audio beep for bad frames: 1
```

Back to DSP Block index

False Color

Description: Maps luminance values to a false-color palette, making intensity differences visually distinctive. Useful for thermal-style visualisation or artistic effect.

Inputs: RGB24 frame (converted to luminance internally)

Outputs: RGB24 false-color frame

Parameters:

- **Palette** — Color palette index. Range: 0–4. Default: 0 (Ironbow).
  - 0 — Ironbow
  - 1 — Rainbow
  - 2 — Greyscale Inverted
  - 3 — Hot
  - 4 — Cool

Example:

```
// Apply a Rainbow palette to a monochrome camera feed
Monochrome → False Color → Palette: 1
```

Back to DSP Block index

Flip Mirror

Description: Flips the frame horizontally, vertically, or both. No pixel values are changed — only the geometric arrangement.

Inputs: RGB24 frame

Outputs: RGB24 geometrically flipped frame

Parameters:

- **Flip horizontal** — Mirror left–right. 0 = off, 1 = on. Default: 0.
- **Flip vertical** — Flip top–bottom. 0 = off, 1 = on. Default: 0.

Example:

```
// Correct a camera mounted upside-down
Flip Mirror → Flip horizontal: 0, Flip vertical: 1
```

Back to DSP Block index

Focusometer

Description: Measures the focus sharpness of an adjustable Region of Interest (ROI) and displays the result as an on-screen meter, text score, and/or audio beep whose pitch tracks focus quality. Useful for manual lens focusing.

Inputs: RGB24 frame

Outputs: RGB24 frame with focus overlay drawn

Parameters:

- **Text** — Show numeric focus score on frame. 0/1. Default: 1.
- **Beep** — Emit a continuous beep whose pitch rises with focus quality. 0/1. Default: 0.
- **Meter** — Show a horizontal bar meter at the top of the frame. 0/1. Default: 1.
- **Show ROI** — Draw the ROI rectangle on the frame. 0/1. Default: 1.
- **ROI %** — Size of the measurement ROI as a percentage of the shorter frame dimension. Range: 20–80. Default: 40.
- **Offset X %** — Horizontal offset of the ROI centre from the frame centre. Range: −100 to 100. Default: 0.
- **Offset Y %** — Vertical offset of the ROI centre from the frame centre. Range: −100 to 100. Default: 0.
- **Noise floor** — Gradient values below this level are ignored. Range: 0–64. Default: 8.
- **Reset** — Button: resets the running focus score statistics.

Example:

```
// Focus aid with audio feedback
Focusometer → ROI%: 30, Beep: 1, Meter: 1, Text: 1
```

Back to DSP Block index

Gain Color

Description: Multiplies each RGB channel by an independent gain factor. Values are scaled around unity at the midpoint (128) of the 0–256 range.

Inputs: RGB24 frame

Outputs: RGB24 frame with per-channel gain applied (clamped to 0–255)

Parameters:

- **Gain red** — Red channel multiplier. Range: 0–256. Default: 128 (unity).
- **Gain green** — Green channel multiplier. Range: 0–256. Default: 128 (unity).
- **Gain blue** — Blue channel multiplier. Range: 0–256. Default: 128 (unity).

Example:

```
// Boost red, reduce blue for a warmer look
Gain Color → Gain red: 160, Gain green: 128, Gain blue: 90
```

Back to DSP Block index

Gain White

Description: Multiplies all three RGB channels by a single gain factor, uniformly brightening or darkening the image.

Inputs: RGB24 frame

Outputs: RGB24 frame with uniform gain applied

Parameters:

- **Gain** — Multiplier applied to all channels. Range: 0–256. Default: 128 (unity).

Example:

```
// Double the brightness
Gain White → Gain: 200
```

Back to DSP Block index

Gamma

Description: Applies a power-law (gamma) tone curve to all channels using a look-up table. Values below 1.0 brighten mid-tones; values above 1.0 darken them.

Inputs: RGB24 frame

Outputs: RGB24 gamma-corrected frame

Parameters:

- **Gamma x10** — Gamma value × 10. Range: 1–50. Default: 10 (gamma = 1.0, linear). Set to 5 for γ = 0.5 (brighter mid-tones).

Example:

```
// Brighten mid-tones with gamma 0.5
Gamma → Gamma x10: 5
```

Back to DSP Block index

Median

Description: Applies a 3×3 median filter to each color channel independently. Highly effective at removing salt-and-pepper (impulse) noise without blurring edges as strongly as a mean blur.

Inputs: RGB24 frame

Outputs: RGB24 median-filtered frame

Parameters: None

Example:

```
// Remove hot pixels from an astronomy camera
Median
```

Back to DSP Block index

Monochrome

Description: Converts the color frame to greyscale using a standard luminance-weighted sum of the RGB channels. The output remains an RGB24 frame with equal R, G, and B values.

Inputs: RGB24 frame

Outputs: RGB24 greyscale frame

Parameters: None

Example:

```
// Convert to greyscale before applying a false-color palette
Monochrome → False Color → Palette: 0
```

Back to DSP Block index

Negative

Description: Inverts all pixel values (output = 255 − input for each channel), producing a photographic negative of the frame.

Inputs: RGB24 frame

Outputs: RGB24 inverted frame

Parameters: None

Example:

```
// Invert after threshold for white-on-black objects
Threshold → Level: 128, Invert: 0
Negative
```

Back to DSP Block index

Noise Color

Description: Overlays independent pseudo-random noise on each RGB channel. The noise amplitude for each channel is controlled independently, allowing coloured noise effects.

Inputs: RGB24 frame

Outputs: RGB24 frame with per-channel noise added

Parameters:

- **Amp red** — Noise amplitude for the red channel. Range: 0–255. Default: 128.
- **Amp green** — Noise amplitude for the green channel. Range: 0–255. Default: 128.
- **Amp blue** — Noise amplitude for the blue channel. Range: 0–255. Default: 128.

Example:

```
// Add low-level color noise for CRT simulation
Noise Color → Amp red: 20, Amp green: 20, Amp blue: 20
```

Back to DSP Block index

Noise White

Description: Overlays uniform (luminance) pseudo-random noise equally on all RGB channels.

Inputs: RGB24 frame

Outputs: RGB24 frame with white noise added

Parameters:

- **Amp white (all RGB)** — Noise amplitude applied equally to R, G, and B. Range: 0–255. Default: 128.

Example:

```
// Simulate sensor noise
Noise White → Amp white: 30
```

Back to DSP Block index

Object Tracker

Description: Detects and tracks moving objects in the scene using a delta-grid algorithm. Unconfirmed detections (candidate blobs) and confirmed entities (tracked objects) are shown as overlays. Generates audio alerts when new activity is detected. Useful for surveillance, wildlife monitoring, or astronomy.

Inputs: RGB24 frame

Outputs: RGB24 frame with tracking overlays drawn

Parameters:

- **Pre stack for delta** — Number of frames to average before computing the delta. Range: 0–10. Reduces noise before detection.
- **Delta noise floor** — Pixel difference values below this level are ignored.
- **MAD Threshold x100** — Sensitivity threshold expressed as a multiple of the Median Absolute Deviation × 100. Range: 50–550. Default: 150 (1.5×MAD).
- **Constant crop enable** — Restrict tracking to a central crop region. 0/1.
- **Constant crop pixels** — Pixel radius of the central crop region. Range: 32–512. Default: 64.
- **Show RGB24 delta frame** — Overlay the raw delta image. 0/1.
- **Show delta grid detections** — Highlight grid cells that exceed the threshold. 0/1.
- **Show unconfirmed entities** — Draw unconfirmed candidate bounding boxes. 0/1.
- **Show confirmed entities** — Draw confirmed tracked entity bounding boxes. 0/1.
- **Show crop bounds** — Draw the constant crop boundary. 0/1.
- **Show entity nonce** — Display each entity's unique identifier. 0/1.
- **Audio alert on new activity** — Play a sound when a new entity is first confirmed. 0/1.
- **Verbose state logging** — Log detailed tracking state changes to the Log page. 0/1.

Example:

```
// Detect moving objects with audio alert
Object Tracker → MAD Threshold x100: 150, Show confirmed entities: 1, Audio alert: 1
```

Back to DSP Block index

Reticule Cartesian

Description: Draws a Cartesian (rectangular) grid reticule on the frame, optionally with axis labels and lens-distortion correction. Useful for aligning telescopes, microscopes, or camera rigs.

Inputs: RGB24 frame

Outputs: RGB24 frame with grid overlay drawn

Parameters:

- **Grid labels** — Show numerical axis labels on the grid lines. 0/1. Default: 1.
- **Opacity %** — Blend opacity of the reticule overlay. Range: 0–100. Default: 50.
- **Pixel thickness** — Line thickness in pixels. Range: 1–8. Default: 1.
- **Grid count X** — Number of vertical grid lines. Range: 2–100. Default: 8.
- **Grid count Y** — Number of horizontal grid lines. Range: 2–100. Default: 8.
- **Offset X x 0.1%** — Horizontal offset of the grid origin. Range: −500 to 500. Default: 0.
- **Offset Y x 0.1%** — Vertical offset of the grid origin. Range: −500 to 500. Default: 0.
- **Scale X %** — Horizontal scale adjustment. Range: −90 to 110. Default: 0.
- **Scale Y %** — Vertical scale adjustment. Range: −90 to 110. Default: 0.
- **Distortion x 0.1%** — Barrel/pincushion lens distortion correction. Range: −500 to 500. Default: 0.
- **Diagonal FOV degrees** — Diagonal field of view of the lens for accurate scaling. Range: 1–180. Default: 45.

Example:

```
// 10×10 grid with 50% opacity for a telescope eyepiece
Reticule Cartesian → Grid count X: 10, Grid count Y: 10, Opacity%: 50, Diagonal FOV: 2
```

Back to DSP Block index

Reticule Polar

Description: Draws a polar (radial) reticule with concentric rings and radial sector lines. Supports the same distortion and FOV correction as the Cartesian reticule.

Inputs: RGB24 frame

Outputs: RGB24 frame with polar grid overlay drawn

Parameters:

- **Grid labels** — Show angular/ring labels. 0/1. Default: 1.
- **Opacity %** — Overlay blend opacity. Range: 0–100. Default: 50.
- **Pixel thickness** — Line thickness in pixels. Range: 1–8. Default: 1.
- **Sectors** — Number of radial sector divisions. Range: 1–50. Default: 12.
- **Rings** — Number of concentric rings. Range: 1–50. Default: 8.
- **Offset X x 0.1%** — Horizontal centre offset. Range: −500 to 500. Default: 0.
- **Offset Y x 0.1%** — Vertical centre offset. Range: −500 to 500. Default: 0.
- **Scale X %** — Horizontal scale. Range: −90 to 110. Default: 0.
- **Scale Y %** — Vertical scale. Range: −90 to 110. Default: 0.
- **Distortion x 0.1%** — Lens distortion correction. Range: −500 to 500. Default: 0.
- **Diagonal FOV degrees** — Lens diagonal FOV for accurate ring spacing. Range: 1–180. Default: 45.

Example:

```
// Polar reticule for a circular telescope eyepiece
Reticule Polar → Sectors: 12, Rings: 6, Opacity%: 40
```

Back to DSP Block index

Save Movie

Description: Records the processed video stream to a movie file at a configurable output frame rate. A new file is created at each interval. Supports audio beep and speech notifications on file creation.

Inputs: RGB24 frame

Outputs: RGB24 frame (pass-through); files written to `<folder>\movies\`

Parameters:

- **Interval seconds** — Duration of each output movie file in seconds. Range: 0–3600. Default: 1. Set to 0 to record a single continuous file.
- **Output FPS** — Output movie frame rate. Range: 1–120. Default: 20.

- **Beep** — Beep when a new file segment starts. 0/1. Default: 1.
- **Speak** — Announce new file segments via TTS. 0/1. Default: 0.
- **Browse folder** — Button: open a folder picker to set the output root directory.

Example:

```
// Record 60-second clips at 25 fps with beep notification
Save Movie → Interval seconds: 60, Output FPS: 25, Beep: 1
```

Back to DSP Block index

Save Photos

Description: Saves individual frames as image files at a configurable time interval. Useful for time-lapse capture. Files are saved to `<folder>\photos\`.

Inputs: RGB24 frame

Outputs: RGB24 frame (pass-through); images written to disk

Parameters:

- **Interval seconds** — Time between successive photo captures. Range: 0–3600. Default: 5. Set to 0 to capture every frame.
- **Beep** — Beep on each capture. 0/1. Default: 1.
- **Speak** — Announce each capture via TTS. 0/1. Default: 0.
- **Browse folder** — Button: set the output root directory.

Example:

```
// Time-lapse: one photo every 30 seconds
Save Photos → Interval seconds: 30, Beep: 1
```

Back to DSP Block index

Sharpen

Description: Applies an unsharp mask to enhance edge contrast and improve perceived sharpness.

Inputs: RGB24 frame

Outputs: RGB24 sharpened frame

Parameters:

- **Amount** — Sharpening strength. Range: 1–100. Default: 50.

Example:

```
// Sharpen after stacking to recover edge detail
Stack → Frame stack count: 16
Sharpen → Amount: 70
```

Back to DSP Block index

Snap Photo

Description: Captures a single frame to disk on demand. The image is saved to a dated sub-folder inside `<root>\snapshots\`. Supports beep and TTS confirmation.

Inputs: RGB24 frame

Outputs: RGB24 frame (pass-through); one image saved on trigger

Parameters:

- **Beep** — Beep when the snapshot is saved. 0/1. Default: 1.
- **Speak** — Announce the capture via TTS. 0/1. Default: 0.
- **Take snapshot** — Button: immediately capture the current frame.
- **Browse snapshots** — Button: open the snapshots folder in File Explorer.

Example:

```
// Manual snapshot at the end of the pipeline
... → Snap Photo → Beep: 1, Take snapshot (button)
```

Back to DSP Block index

Spatial Correlator

Description: Stabilises video by computing the 2D spatial cross-correlation between the current frame and a stored anchor frame, then translating the current frame to align it with the anchor. Useful for compensating for camera vibration or drift.

Inputs: RGB24 frame

Outputs: RGB24 frame translated to align with the anchor

Parameters:

- **Reset anchor threshold** — Automatically reset the anchor frame when the alignment offset exceeds this percentage. Prevents runaway drift.
- **Wrapping** — Use wrapping (toroidal) translation instead of cropping at the edges. 0/1.
- **Reset anchor frame** — Button: manually capture the current frame as the new alignment anchor.

Example:

```
// Stabilise a drifting telescope feed
Spatial Correlator → Wrapping: 0, Reset anchor frame (button)
```

Back to DSP Block index

Sprite Orbs

Description: Overlays animated glowing orb sprites that fly across the frame in random directions. The orbs have a soft radial gradient. Supports a spherical-remap mode that makes orbs appear to travel over a sphere.

Inputs: RGB24 frame

Outputs: RGB24 frame with animated orbs drawn on top

Parameters:

- **Speed** — Base orb travel speed (1–100 mapped to 20–2000 px/s). Default: 25.
- **Speed random** — Randomness added to each orb's speed. Default: 25.
- **Size** — Base orb diameter (1–100 mapped to 1–100 px). Default: 25.
- **Size random** — Randomness added to each orb's diameter. Default: 25.
- **Occurrence** — Rate at which new orbs are spawned. Default: 25.
- **Spheric** — Enable spherical perspective remapping. 0/1. Default: 0.

Example:

```
// Bokeh-like orb overlay at low occurrence
Sprite Orbs → Speed: 15, Size: 40, Occurrence: 10, Spheric: 0
```

Back to DSP Block index

Sprite Squares

Description: Overlays a configurable number of animated bouncing character-glyph sprites that move around the frame and bounce off the edges.

Inputs: RGB24 frame

Outputs: RGB24 frame with animated square sprites drawn on top

Parameters:

- **Sprites** — Number of animated sprites. Range: 0–255. Default: 7.

Example:

```
// Overlay 20 bouncing sprites
Sprite Squares → Sprites: 20
```

Back to DSP Block index

Stack

Description: Accumulates multiple frames into a running average, effectively reducing random noise by √N for N stacked frames. Particularly useful for astronomy and low-light imaging. Processing throughput is significantly higher than comparable tools (up to 250 fps at 1080p).

Inputs: RGB24 frame

Outputs: RGB24 averaged frame

Parameters:

- **Frame stack count** — Number of frames to average. Range: 0–255. Default: 16. Set to 0 to disable stacking (pass-through).

Example:

```
// Stack 32 frames to reduce noise by ~5.7×
Stack → Frame stack count: 32
```

Back to DSP Block index

Stream H264

Description: Encodes the processed frame stream as H.264 and serves it over an HTTP live-streaming server. Viewers on the local network can connect with any modern browser or media player.

Inputs: RGB24 frame

Outputs: RGB24 frame (pass-through); H.264 bitstream served over HTTP

Parameters:

- **Port** — TCP port number for the HTTP server. Default: 8080.
- **Open browser** — Button: open the stream URL in the default browser.
- **Copy URL** — Button: copy the stream URL to the clipboard.
- **Restart server** — Button: stop and restart the HTTP server.
- **Port info** — Read-only label showing the current server status and URL.

Example:

```
 // Stream on default port 8080
Stream H264 → Port: 8080, Open browser (button)
// Browser URL: http://<host-ip>:8080
```

Back to DSP Block index

Stream MJPEG

Description: Serves the processed frame stream as a Motion JPEG (MJPEG) HTTP stream. Compatible with all browsers and many IP camera viewers.

Inputs: RGB24 frame

Outputs: RGB24 frame (pass-through); MJPEG stream served over HTTP

Parameters:

- **Port** — TCP port number. Default: 8081.
- **Open browser** — Button: open the stream URL in the default browser.
- **Copy URL** — Button: copy the URL to the clipboard.
- **Restart server** — Button: restart the HTTP server.
- **Port info** — Read-only server status label.

Example:

```
 // MJPEG stream on port 8081
Stream MJPEG → Port: 8081, Open browser (button)
```

Back to DSP Block index

Subtract Frame

Description: Subtracts a previously grabbed reference frame from every subsequent incoming frame. The result highlights differences relative to the reference. Useful for removing fixed-pattern noise, background subtraction, or dark-frame correction.

Inputs: RGB24 frame

Outputs: RGB24 difference frame (current minus reference, clamped to 0–255)

Parameters:

- **Grab frame** — Button: capture the current frame as the reference to subtract from all subsequent frames.

Example:

```
 // Dark frame subtraction for astronomy
// 1. Cover the telescope, click "Grab frame" to capture the dark frame.
// 2. Uncover the telescope to view the dark-subtracted output.
Subtract Frame → Grab frame (button)
```

Back to DSP Block index

Text FPS

Description: Overlays the live pipeline frame rate (frames per second) as a text label on the frame. The FPS value is smoothed with an IIR filter.

Inputs: RGB24 frame

Outputs: RGB24 frame with FPS text drawn

Parameters:

- **Size** — Text scaling factor.
- **Position** — Text anchor position (e.g. top-left, bottom-right). Default: bottom-left.
- **Pixel offset X** — Fine horizontal offset of the text in pixels.
- **Pixel offset Y** — Fine vertical offset of the text in pixels.

Example:

```
// Show FPS in the bottom-left corner
Text FPS → Position: Bottom Left, Size: 1
```

Back to DSP Block index

Text Frame Index

Description: Overlays the running camera frame counter (the total number of frames captured since the camera was connected) on the frame.

Inputs: RGB24 frame

Outputs: RGB24 frame with frame counter text drawn

Parameters:

- **Size** — Text scaling factor.
- **Position** — Text anchor position. Default: top-left.
- **Pixel offset X** — Fine horizontal offset in pixels.
- **Pixel offset Y** — Fine vertical offset in pixels.

Example:

```
// Show frame index in the top-left corner
Text Frame Index → Position: Top Left
```

Back to DSP Block index

Text DSP Index

Description: Overlays the DSP pipeline frame index (the number of frames processed by the pipeline since startup) on the frame.

Inputs: RGB24 frame

Outputs: RGB24 frame with DSP index text drawn

Parameters:

- **Size** — Text scaling factor.
- **Position** — Text anchor position. Default: top-right.
- **Pixel offset X** — Fine horizontal offset in pixels.
- **Pixel offset Y** — Fine vertical offset in pixels.

Example:

```
// Show DSP index in the top-right corner
Text DSP Index → Position: Top Right
```

Back to DSP Block index

Text String

Description: Overlays a user-defined static text string on the frame. Useful for labelling or annotating output video.

Inputs: RGB24 frame

Outputs: RGB24 frame with custom text drawn

Parameters:

- **Size** — Text scaling factor.
- **Position** — Text anchor position. Default: top-center.
- **Pixel offset X** — Fine horizontal offset in pixels.
- **Pixel offset Y** — Fine vertical offset in pixels.
- **Text** — The string to display. Default: "Your text here".

Example:

```
// Watermark the stream with a label
Text String → Text: "Observatory North — 2025", Position: Bottom Center
```

Back to DSP Block index

Text Time

Description: Overlays the current system date and time on the frame as a timestamp. The format follows the system locale.

Inputs: RGB24 frame

Outputs: RGB24 frame with timestamp text drawn

Parameters:

- **Size** — Text scaling factor.
- **Position** — Text anchor position. Default: bottom-center.
- **Pixel offset X** — Fine horizontal offset in pixels.
- **Pixel offset Y** — Fine vertical offset in pixels.

Example:

```
// Add a timestamp to a time-lapse recording
Text Time → Position: Bottom Center
Save Photos → Interval seconds: 60
```

Back to DSP Block index

Threshold

Description: Binarizes the image: pixels with luminance at or above the threshold level are set to white (255); pixels below are set to black (0). An optional invert flag reverses the output polarity.

Inputs: RGB24 frame

Outputs: RGB24 binary (black-and-white) frame

Parameters:

- **Level** — Luminance threshold value. Range: 0–255. Default: 128.
- **Invert** — Invert the binary output. 0 = white above threshold, 1 = black above threshold. Default: 0.

Example:

```
// Segment bright objects from a dark background
Threshold → Level: 200, Invert: 0
```

Back to DSP Block index

Translate

Description: Shifts the frame by a fixed number of pixels in the X and/or Y direction. Can wrap (toroidal shift) or fill vacated pixels with black.

Inputs: RGB24 frame

Outputs: RGB24 translated frame

Parameters:

- **Translate X** — Horizontal pixel shift. Range: −1000 to 1000. Default: 0. Positive values shift right.
- **Translate Y** — Vertical pixel shift. Range: −1000 to 1000. Default: 0. Positive values shift down.
- **Wrapping** — Pixels that exit one edge re-enter from the opposite edge. 0/1. Default: 0.

Example:

```
// Centre-shift a frame that is offset by 50 pixels right
Translate → Translate X: -50, Translate Y: 0, Wrapping: 0
```

Back to DSP Block index

Zebra

Description: Highlights over-exposed and under-exposed pixels with a zebra-stripe pattern. Pixels above the High threshold are drawn as pure red animated stripes; pixels at or below the Low threshold are drawn as pure blue animated stripes. All other pixels pass through unchanged. This helps identify clipping in highlights and crushed shadows.

Inputs: RGB24 frame

Outputs: RGB24 frame with zebra highlighting applied

Parameters:

- **High** — Upper luminance threshold. Pixels at or above this value are clipped to red stripes. Range: 0–255. Default: 250.
- **Low** — Lower luminance threshold. Pixels at or below this value are clipped to blue stripes. Range: 0–255. Default: 0.

Example:

```
// Warn on highlights above 240 and shadows below 10
Zebra → High: 240, Low: 10
```

Back to DSP Block index

5. Pipeline Examples

The following examples show common end-to-end pipeline configurations. Each block is listed in the order it should appear in the pipeline on the Process page.

Low-Light / Astronomy Frame Stacking

Goal: Reduce random sensor noise from a low-light or astronomy camera and record the improved result.

Blocks used: Median → Stack → Sharpen → Text Time → Save Photos

```
 Median                       // Remove hot pixels
Stack      → Frame stack count: 32    // Average 32 frames (~5.7× noise reduction)
Sharpen    → Amount: 60               // Recover edge detail lost in averaging
Text Time  → Position: Bottom Center  // Embed a timestamp
Save Photos → Interval seconds: 5    // Save one photo every 5 seconds
```

Motion Detection with Audio Alert

Goal: Detect moving objects in a static scene, visualise them, and trigger an audio alert.

Blocks used: Blur → Object Tracker → Text Time → Stream MJPEG

```
 Blur           → Radius: 3             // Reduce noise before detection
Object Tracker → MAD Threshold x100: 150
                 Show confirmed entities: 1
                 Audio alert on new activity: 1
Text Time      → Position: Bottom Center
Stream MJPEG   → Port: 8081            // Monitor remotely in a browser
```

Time-Lapse Recording

Goal: Capture a time-lapse sequence with a visible timestamp overlay.

Blocks used: Gain White → Gamma → Text Time → Text String → Save Photos

```
 Gain White  → Gain: 140                    // Slightly boost exposure
Gamma        → Gamma x10: 8                 // Mild gamma lift for mid-tones
Text Time    → Position: Bottom Center      // Live timestamp
Text String → Text: "Garden Cam — 2025"
              Position: Top Center
Save Photos → Interval seconds: 30          // One photo every 30 s
```

Focus Aid for Manual Lenses

Goal: Assist manual focusing with both a visual meter and audio feedback, rejecting blurry frames.

Blocks used: Focusometer → Discriminator

```
 Focusometer  → ROI%: 40, Beep: 1, Meter: 1, Text: 1, Show ROI: 1
Discriminator → Focus selectivity: 50      // Accept top 50% of frames by sharpness
                Audio beep for bad frames: 0
```

Processed Network Streaming

Goal: Apply colour correction, add a watermark, and stream the result over the local network.

Blocks used: Gain Color → Gamma → Text String → Stream H264

```
 Gain Color  → Gain red: 140, Gain green: 128, Gain blue: 110
Gamma       → Gamma x10: 9
Text String → Text: "Live Feed", Position: Top Center
Stream H264 → Port: 8080          // View at http://<host-ip>:8080
```

Dark Frame Subtraction

Goal: Remove fixed-pattern sensor noise (dark frame) from an astronomy feed.

Blocks used: Subtract Frame → Stack → Save Movie

```
 // Step 1: Cover the telescope objective, click "Grab frame" in Subtract Frame.
// Step 2: Uncover the objective.
Subtract Frame → Grab frame (button)
Stack          → Frame stack count: 16
Save Movie     → Interval seconds: 300, Output FPS: 25   // 5-minute clips
```

False-Color Thermal-Style View

Goal: Convert a monochrome camera feed to a thermal-style false-color display.

Blocks used: Monochrome → Gain White → Gamma → False Color

```
 Monochrome  // Convert color camera to greyscale
Gain White  → Gain: 150              // Boost overall brightness
Gamma       → Gamma x10: 7           // Lift mid-tones
False Color → Palette: 0             // Ironbow palette
```

Planetary video frame stacking

Goal: Improve the quality of a planetary video by stacking the best frames.

Blocks used: Discriminator → Spatial correlataor -> Stack → Sharpen → Text Time → Save Movie

```
Discriminator → Sharpness selectivity: 30, Show metric
Spatial Correlator → Wrapping: 1, Reset anchor frame (button)
Stack → Frame stack count: 64
Sharpen → Amount: 80
Text Time → Position: Bottom Center
Save Movie → Interval seconds: 60, Output FPS: 25, Beep: 1
```

6. Troubleshooting & FAQs

Common Issues

> **Issue:** Something isn't working.
> **Solution:** Steps to resolve it.

FAQs

Q: Frequently asked question?
A: Answer.