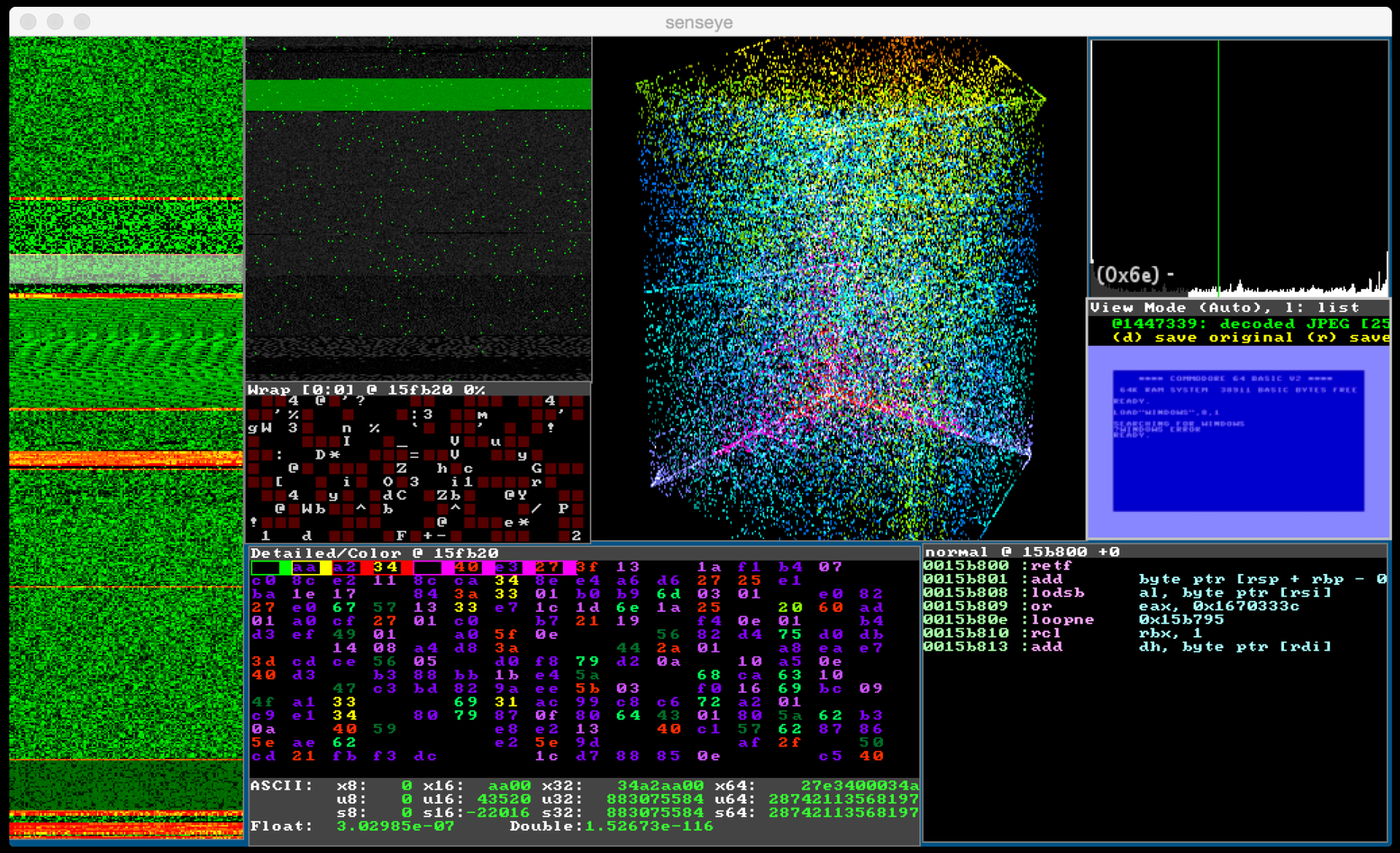
Senseye



IRC #arcan @ irc.freenode.net

Github github.com/letoram/senseye

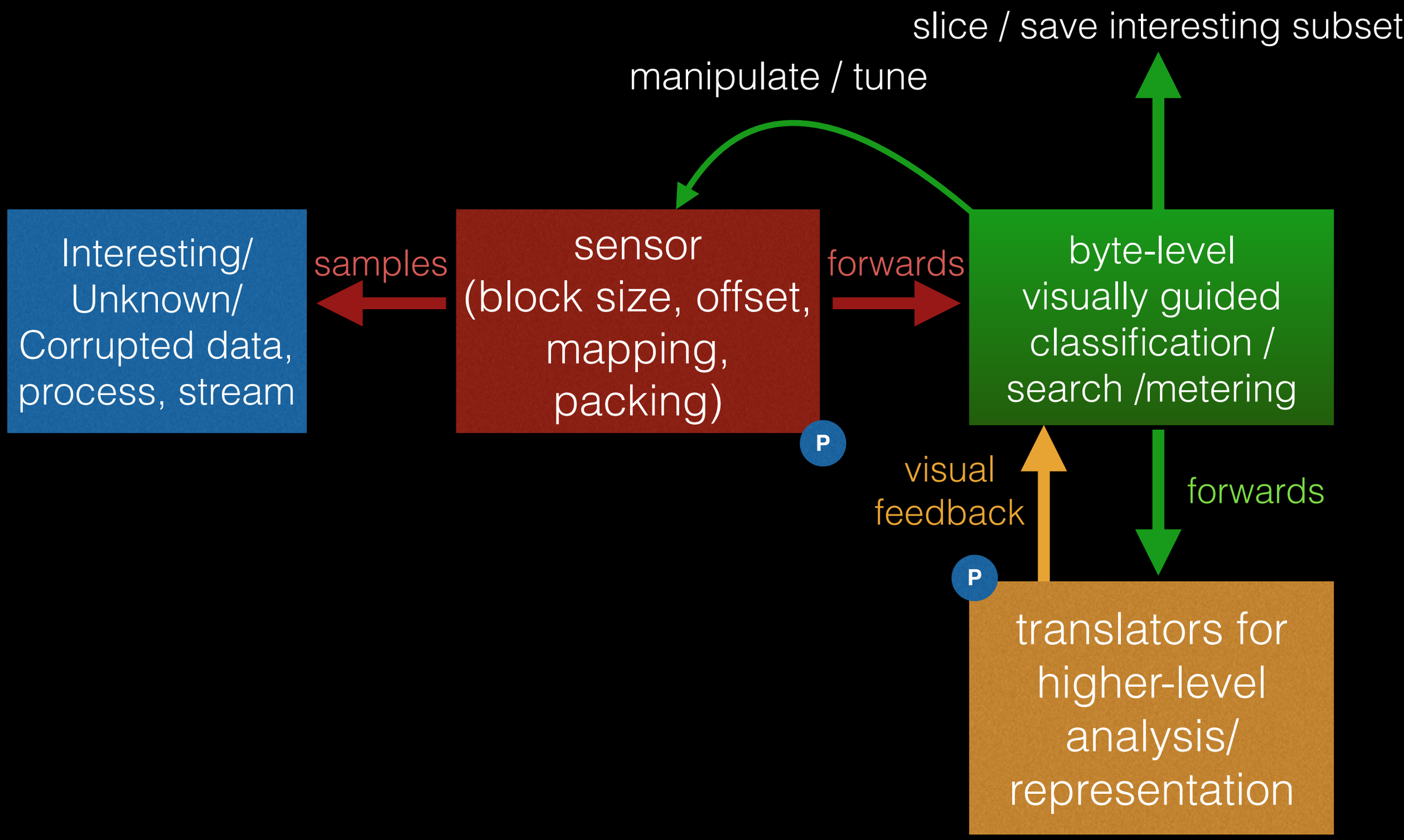
Outline

- Purpose
- Workflow
- Features
  - senses (memory, files, pipes, ...)
  - tools (visualizaion, statistics, searching, specialized)
  - translators (disassembly, hex, ascii, images)
- New features in 0.3 (overlays, fault injection, ...)
- Future Plans
- References

Purpose

- Primarily an human-assistive data analysis tool (in contrast to automated ones).
- Solving ‘needle in haystack’ manual search style problems: e.g. crash dump analysis, debugging, forensics, reverse engineering.
- Finding and exposing hidden structures, data corruption etc. in larger data flows (hundreds of megabytes to gigabytes) than would be feasible with other tools.
- Experiment platform for discovering new data visualization and analysis techniques, to later incorporate in reports and automated tools.

Example Workflow

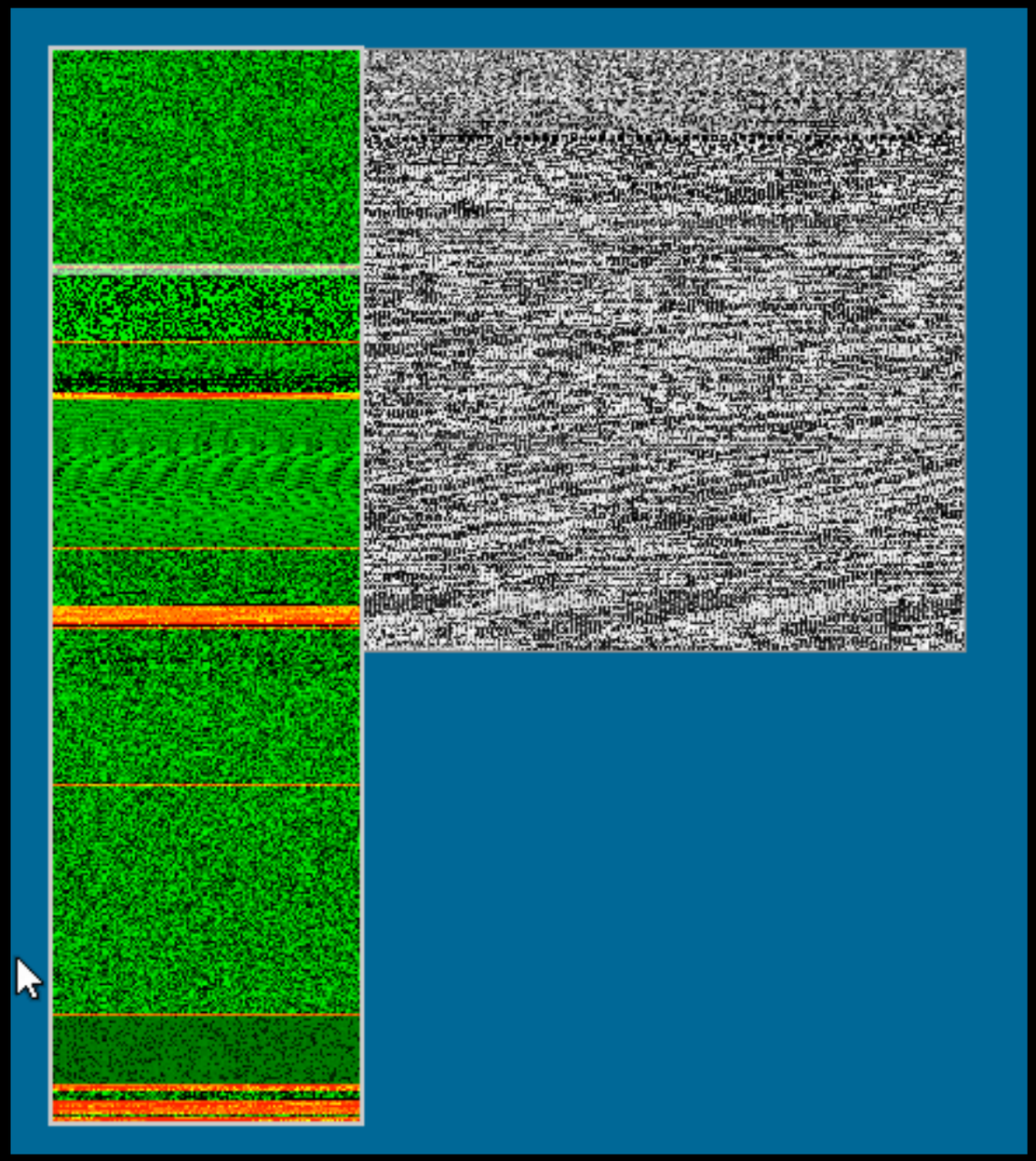


Senses <File>

- Takes one input file (up to a few Gb is reasonable) and progressively steps through

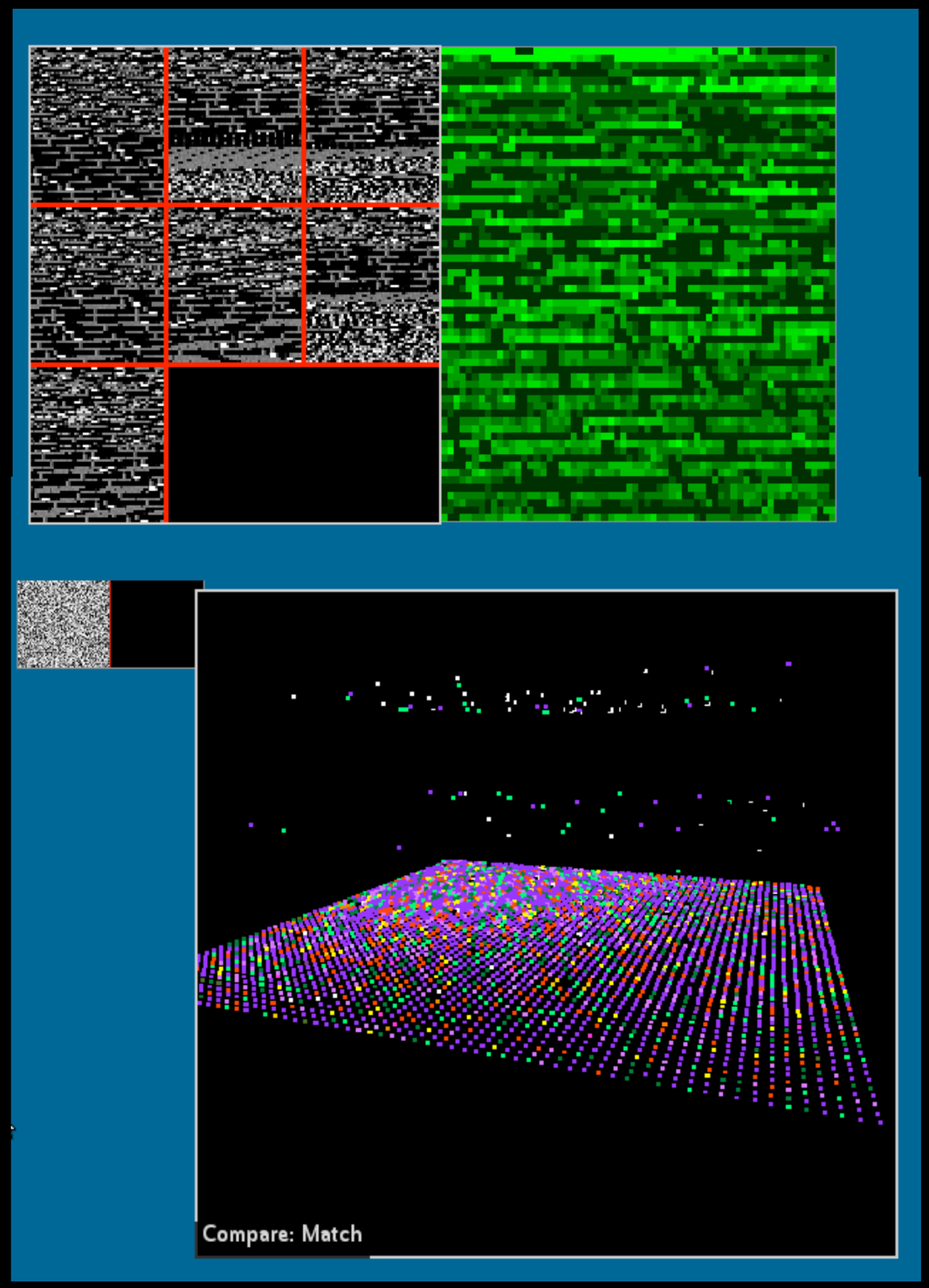
- Navigation window for quick stepping

- Can highlight parts with statistically significant deviations



Senses < MFile >

- Takes multiple input files of suspected same type, for comparison, identification of headers / subheaders / length fields.
- Tiles can be stepped individually
- Metatiles with additional properties, i.e. tile[0]^tile[1]
- 3D diff view

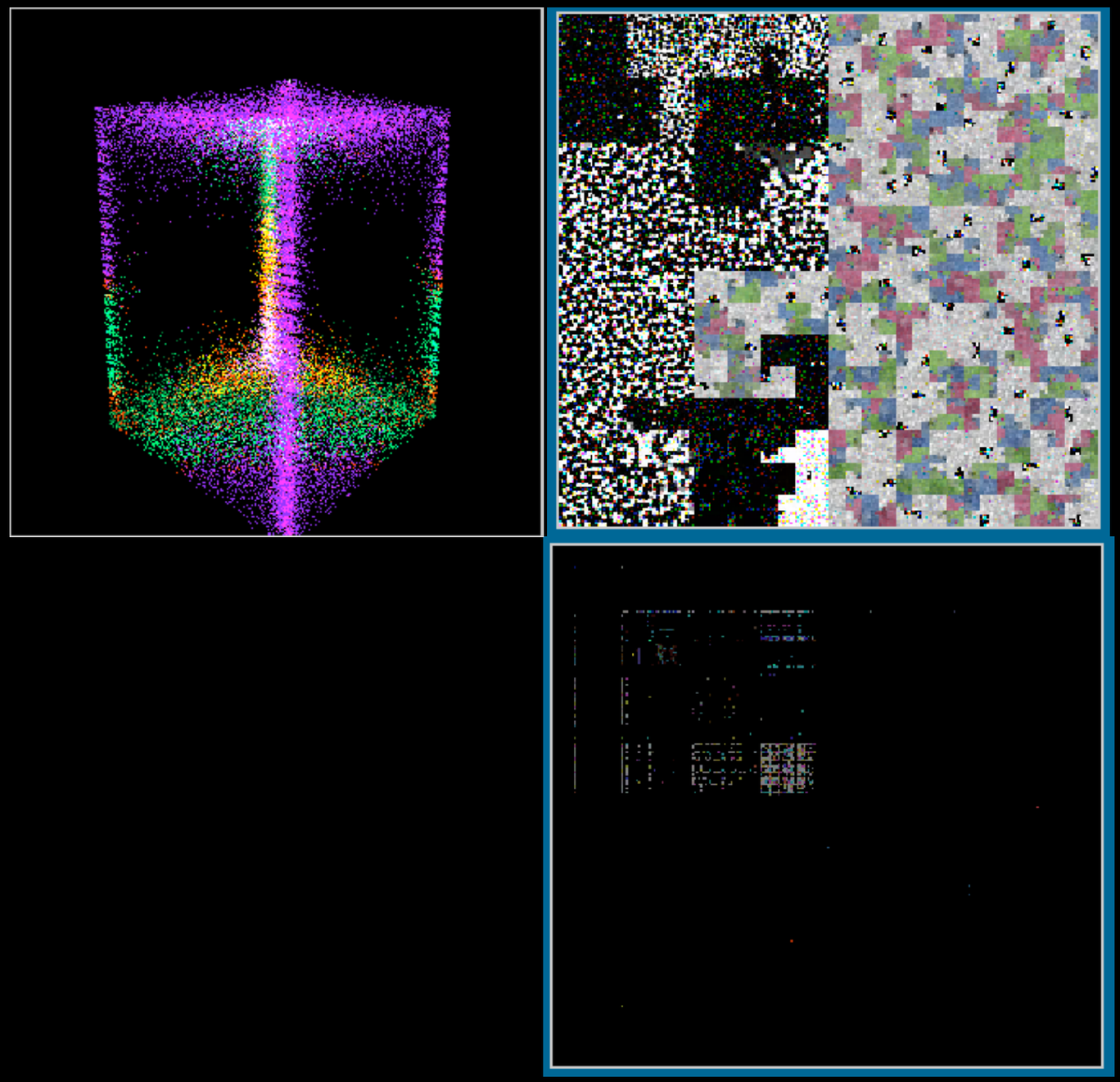


Senses <Mem, Pipe>

- (mem) samples live memory navigating using mapped pages
  - can sample same addr periodically
- (pipe) for use in streaming data (pf redirect rule or cat:ing raw devices)

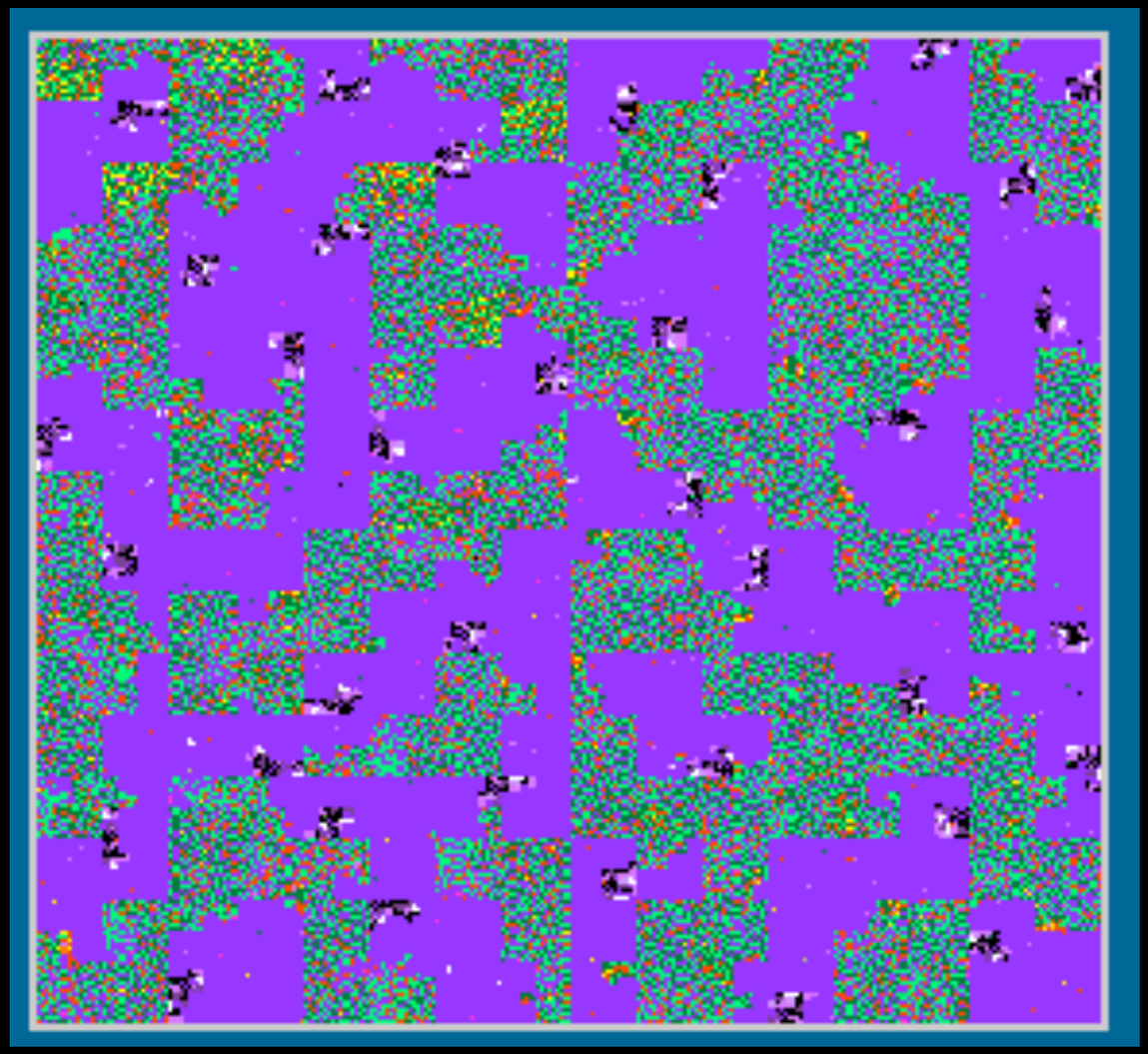
Tools <mapping>

- Mapping modes (zigzag, hilbert, bigram (“tuple”), 3D)
- projections that highlight specific properties (e.g. value clustering, spatial locality, …)



Tools <coloring>

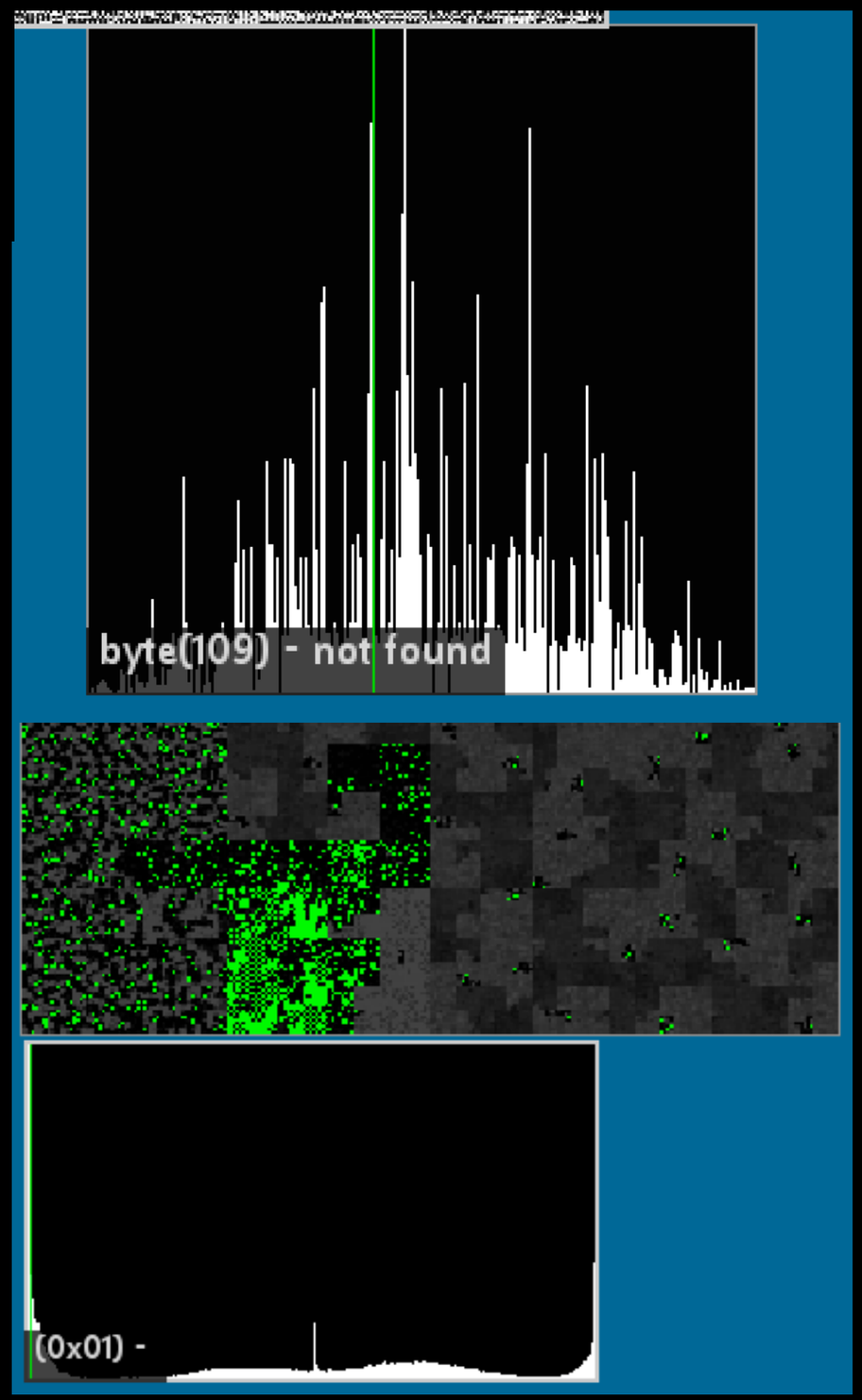
- For highlighting specific values and ranges
- Byte-value used as index into LUT (“palette”)
- Can also be GLSL shaders, so possible to color more than single values



Tools <measuring>

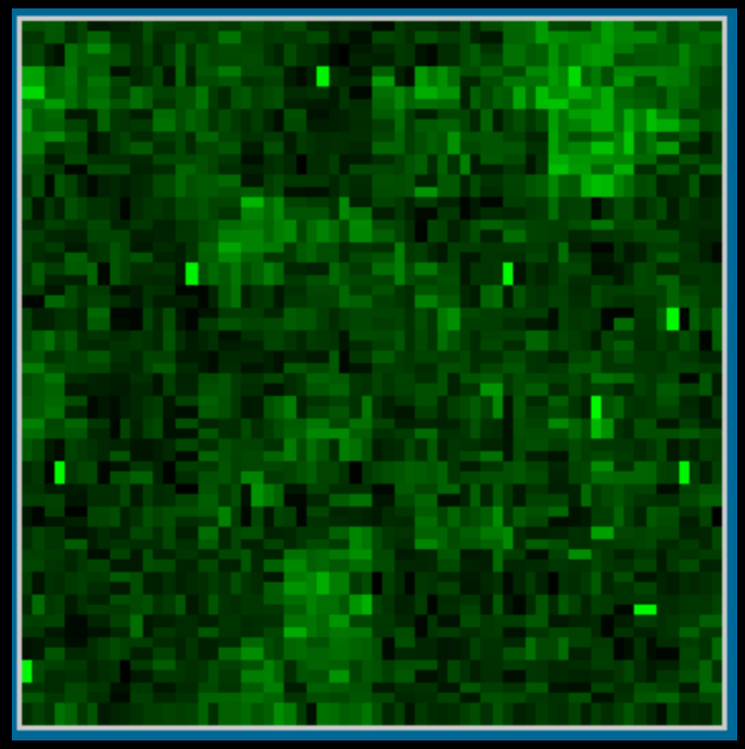
- Byte distance for number of bytes until selected value reoccurs given reference point

- Histogram for byte value frequency and highlighting for distribution



Tools <metadata>

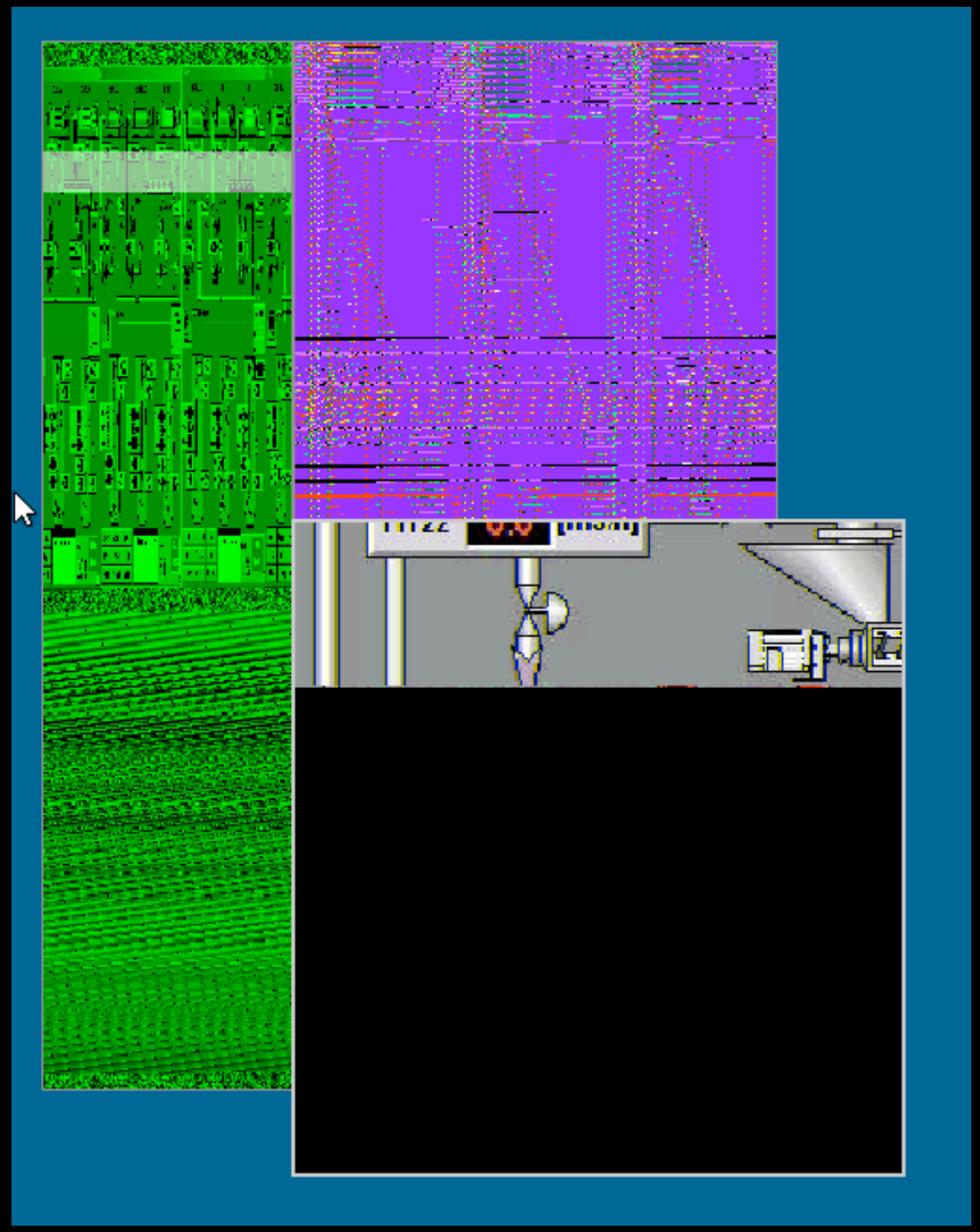
- Metadata from sensor, e.g. entropy, byte-pattern matching for finding compressed / encrypted data, or changes in value between samples (useful for sense_mem)



Tools <special>

- pattern-match search using current data window (works well with projections e.g. bigram) or histogram as reference

- Pict-tuner for manually or automatically detecting stride and colorspace from raw image buffers

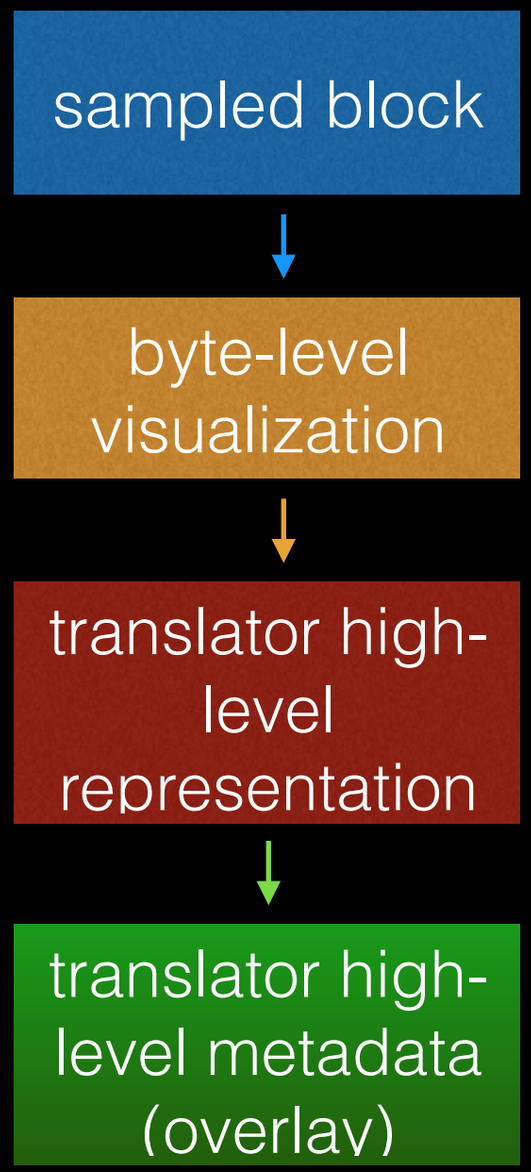


Translators

- Translators provide a higher-level abstraction view of selected subsets

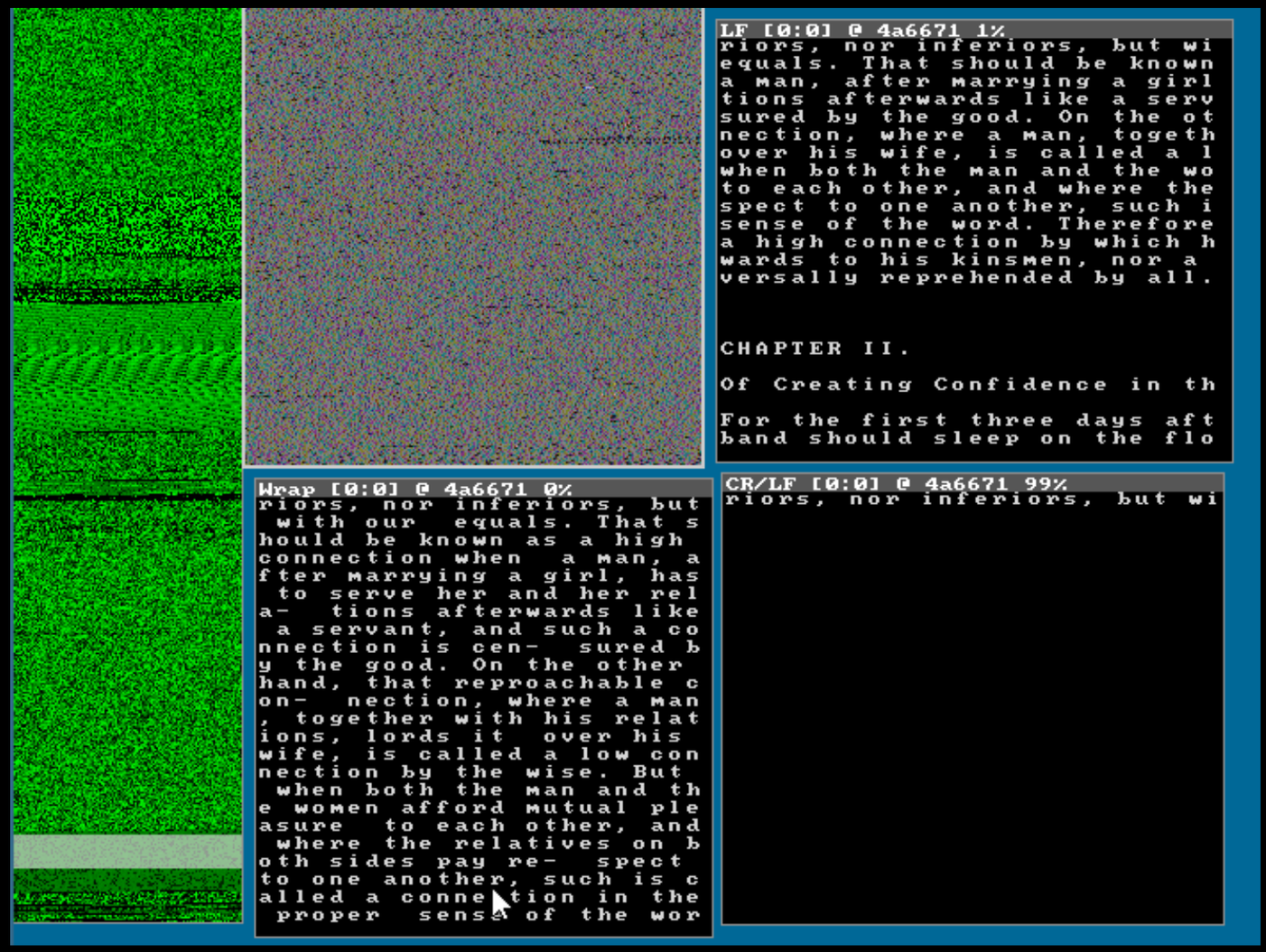
- Current available: hex, ascii, capstone, img, pipe (use with file- or script based classifier)

- parser + visual_representation = translator



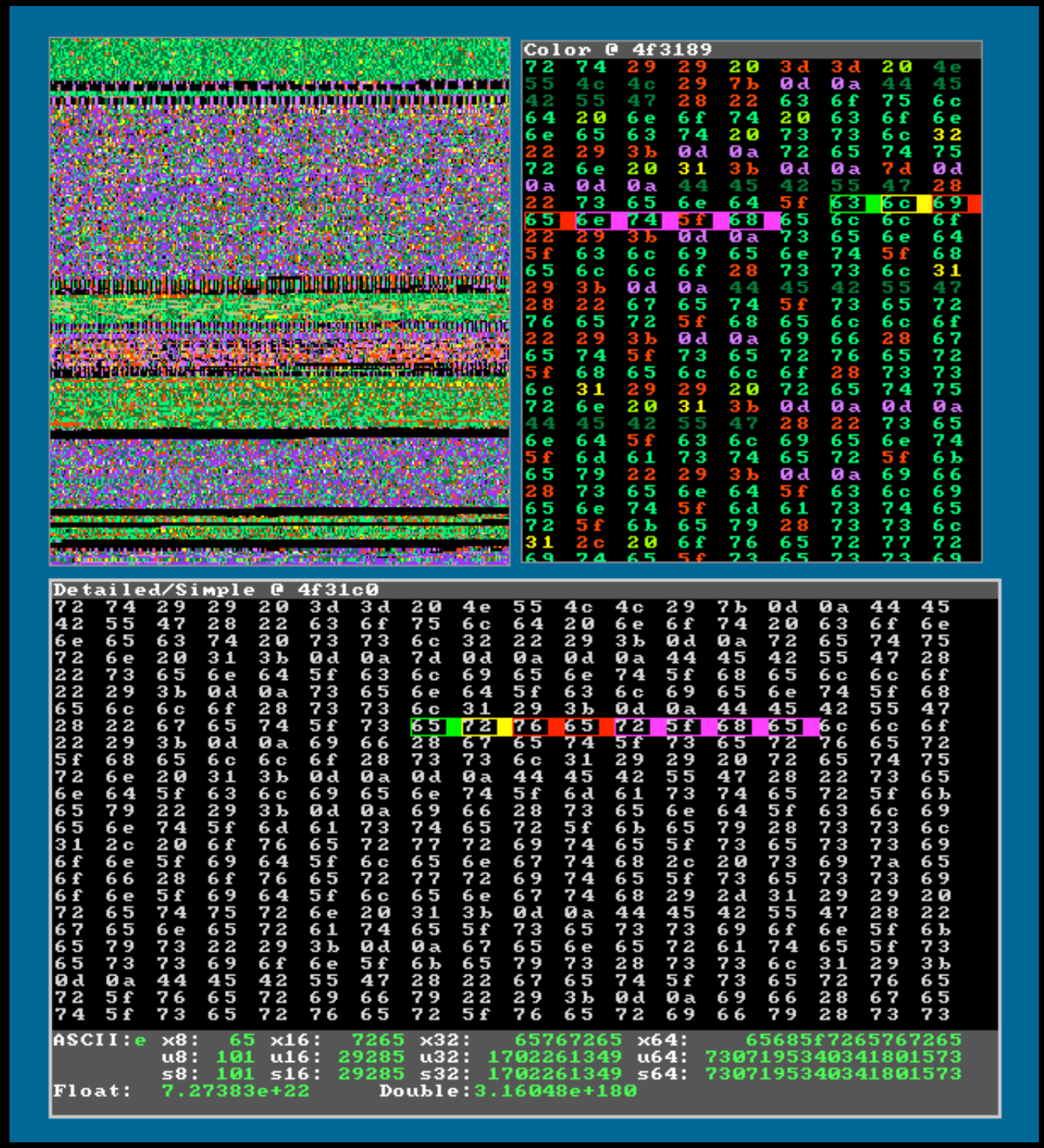
Translator<ascii>

- Simplest possible: 7-bit ascii decoding with three different line-feed modes (LF/WRAP/CR-LF)



Translator<hex>

- numeric / hex view showing a number of byte decoding schemes



Translator<disassembly>

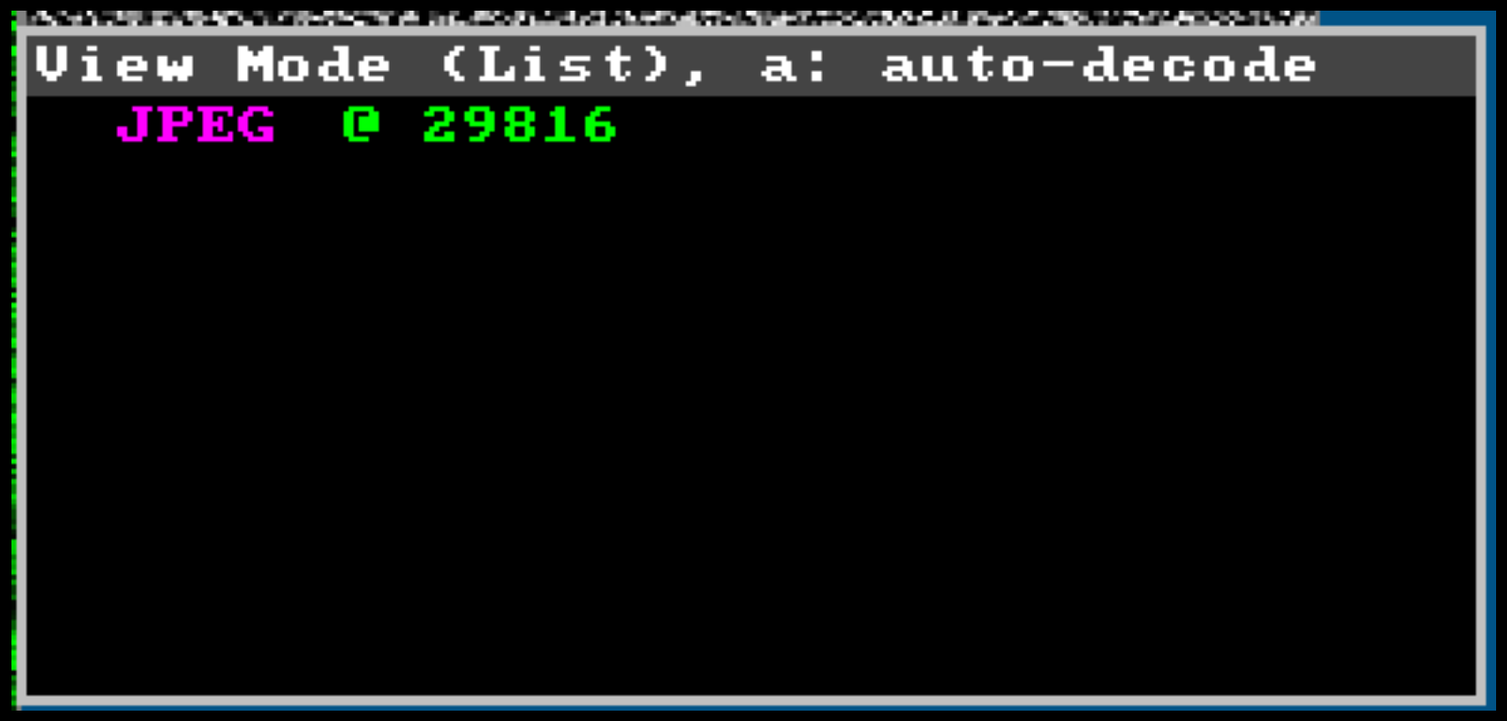
- multi- architecture disassembly based on capstone (but should be trivial to hook up other disassembly engines for side-by-side comparison)

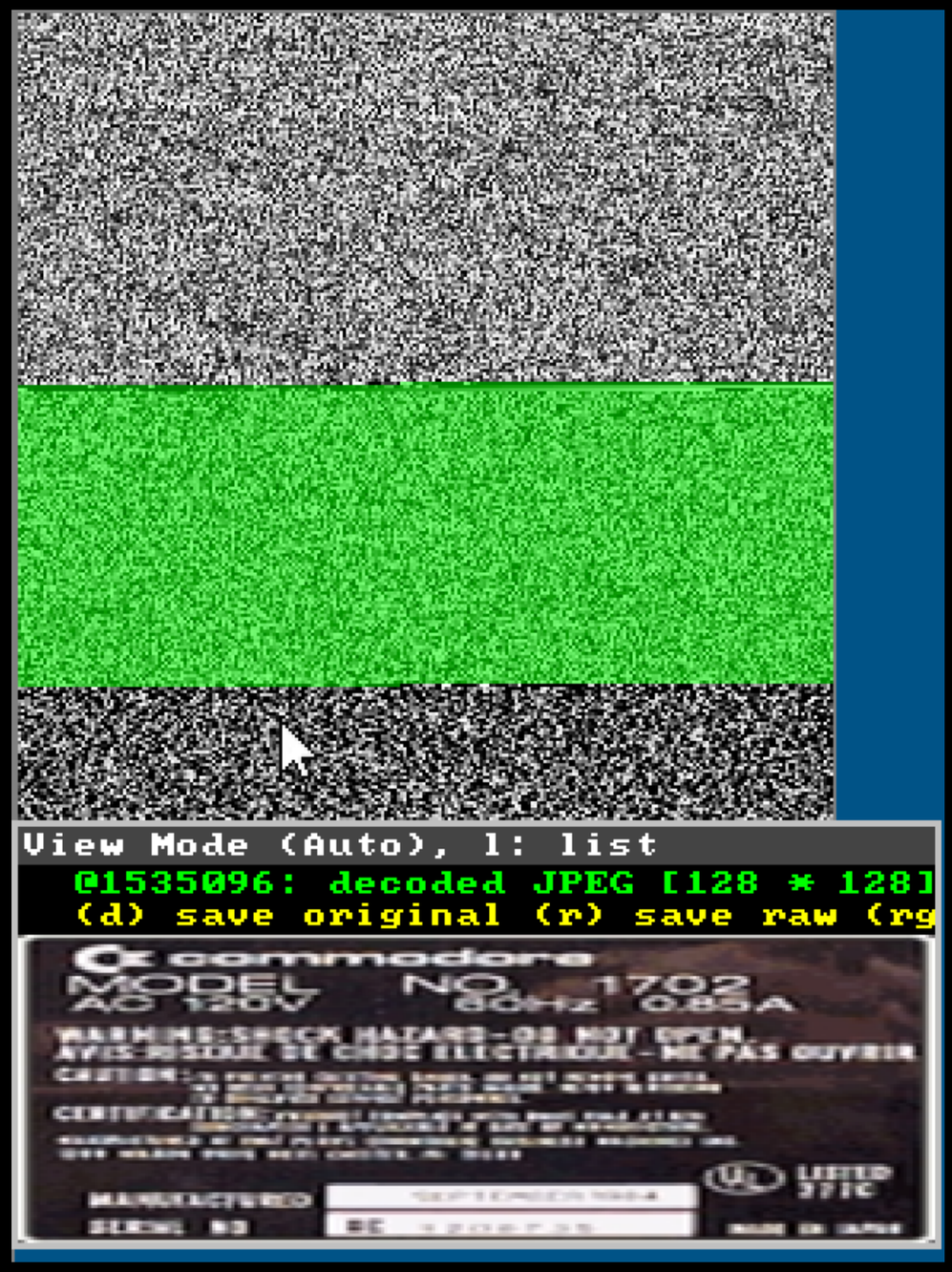
- architecture, output str etc. command line arguments with user defined format string.

- instruction group based coloring

Translator<image>

- Using stb_image parser with magic value based header detection
- modes for showing possible candidates or automatic decoding



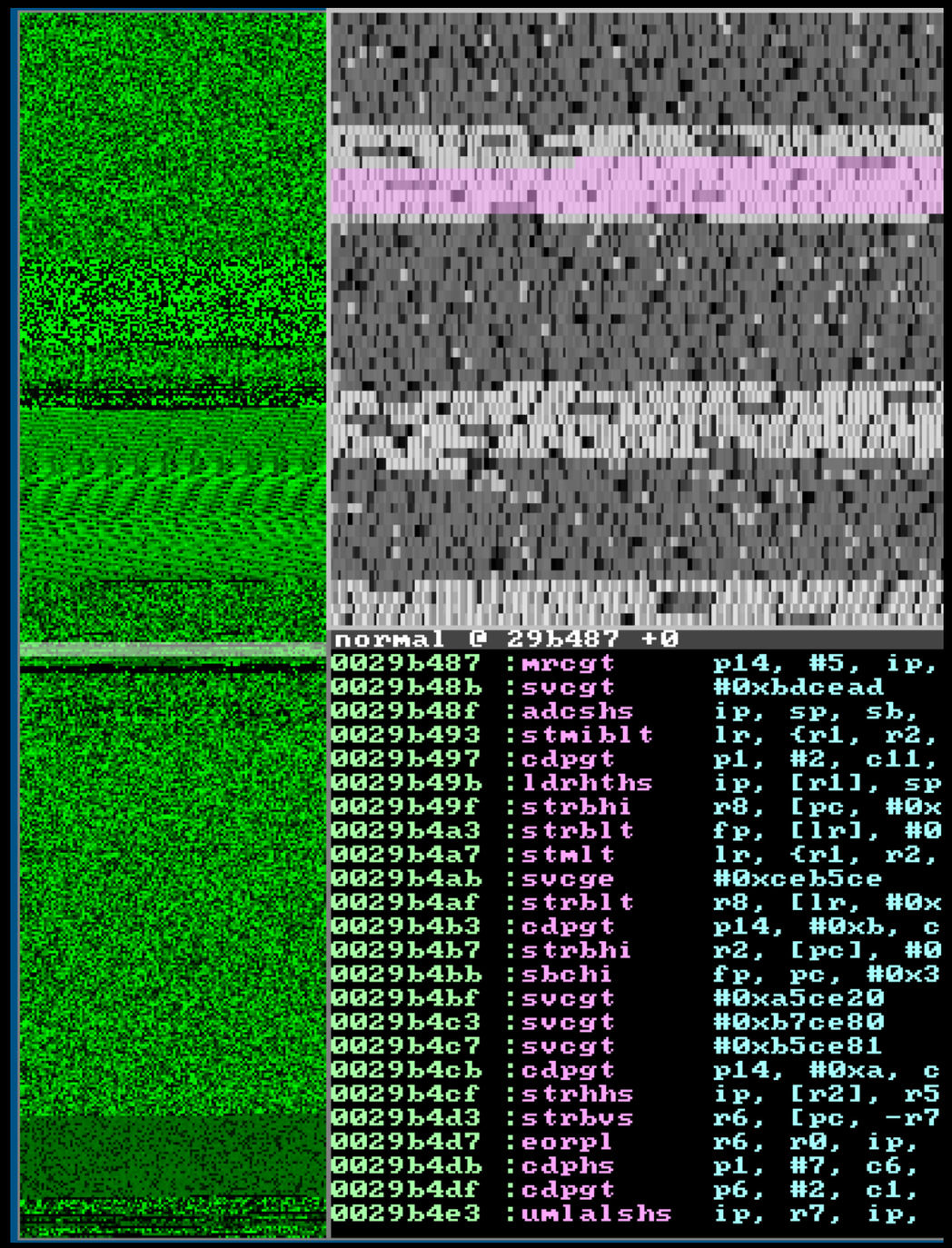


New in 0.3

- sense_mem support for OS X (courtesy of p0sixninja)
- sense_file gets histogram edge highlight in preview
- sense_mfile 3d view / per cell stepping / meta tile (xor, and, …)
- xlt_img - image decoding translator
- overlays (next slides)
- visually guided f&f (next slides)
- translator reconnection on crash
- tuple split to tuple->pack (distr only) / tuple->acc (density)

Overlays

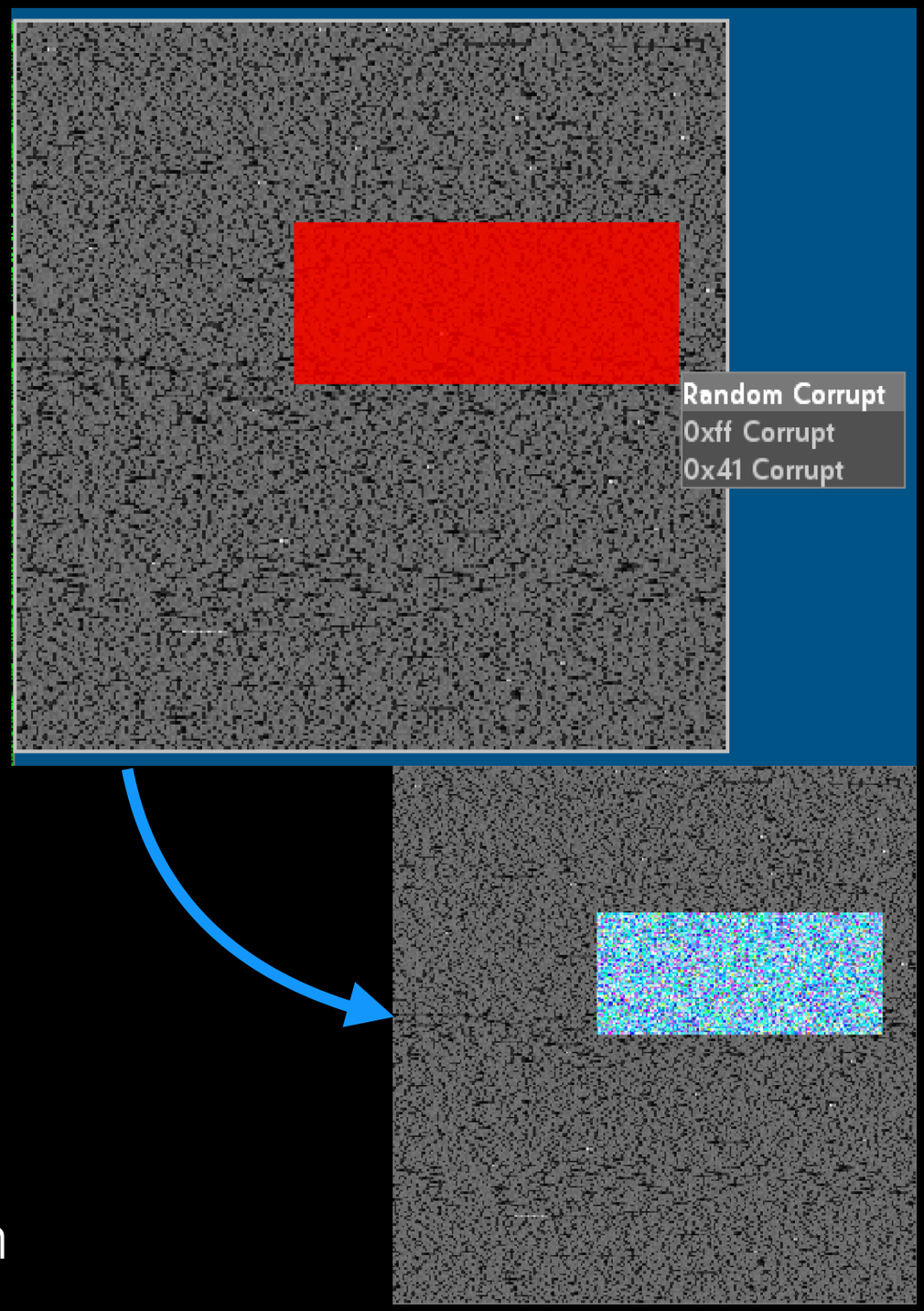
- exposes translator state as a subwindow overlay on the data window
- typically indicates bytes consumed, but can also write detailed data (e.g. symbol names at certain addresses)
- other use-cases would be *corkami*-style field- coloring, highlighting known structures etc.
- alignment has slight precision / synch issues :'(



(fuzzing and fault injection)

Visually Guided F&F

- UI: click (once) or meta+mouse-motion (continuous) in data view will change parsing offset in translators.
- Playback (sliding window at configured step sz) will also change parsing offset and cutoff (window size)
- setup: wrap targeted parser in translator api (like with xlt_img), while(true) { xlt_img; save core; }
- drag-zoom + tab will change state to inject, sensor will manipulate data source (sensor specific) or sampled output
- manipulated sample will be pushed and forwarded to translators that (hopefully) crash on the new input :)



Future Plans

- Next Release (0.4):
  - Serious UI Overhaul (too much drag/resize/keybinds work)
  - Load searching references (histogram/map projection) from file
  - Integrators (sensor+translator in one)
    - Debug integrator (IDA, ECFS, LLDB, R2 …) backends
      - Overlay symbol- data, ida, set triggers (watchpoint)
- xlt_capstone:
  - Add unicorn-engine CPU emulator and UI for setting register states, using sensor backed memory view and debug- controls to run/step.
  - Basic statistics (group/instruction histogram) + instruction helper lookup database

Future Plans

- Later (0.5+)
  - sense_mem (windows support), process controls, huge metapage, refresh trigger (e.g. mprotect)
  - fault injector - user-editable sequences, automated / repeated mutation at selected injection sites
  - PE/ELF Translator
  - sense_file - decompression support, multiple parallel windows, I/O access replay
  - xlt_audio - playback and visualization of raw and compressed (FFT- etc.)
  - Execution tracing (ftrace- like sampling with 3d graph view)

References

- Great Presentations
  - greg conti, sergey bratus, “voyage of the reverser” @ blackhat 2010
  - chris domas, “The Future of RE: Dynamic Binary Visualization” @ recon 2013
- Dated Academics
  - “Retooling and Securing Systemic Debugging” @ nordsec 2012