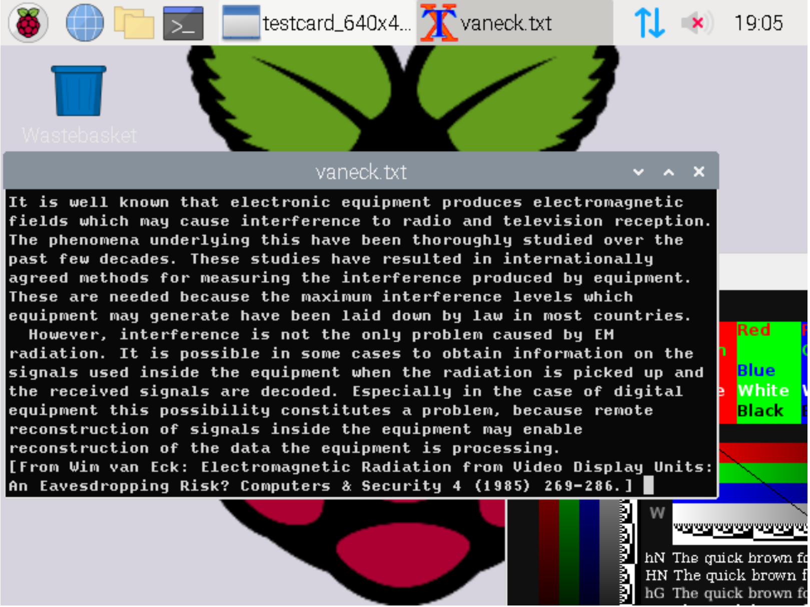
Multiband Pixel Colour Classification from HDMI Emissions

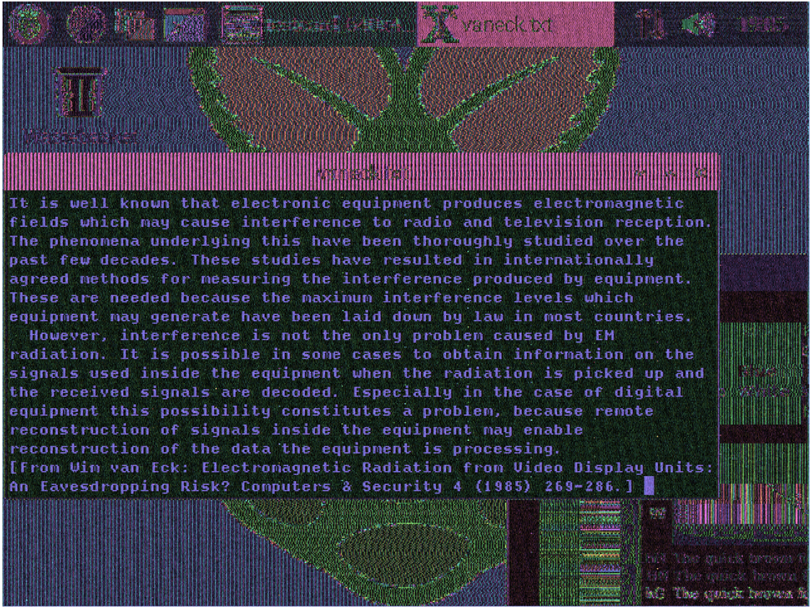
Dimitrije Erdeljan, Markus G. Kuhn

Department of Computer Science and Technology
University of Cambridge

EMC EUROPE 2025



HDMI test image: 640 × 480 pixels (800 × 525 total), $f_v \approx 60.0$ Hz, $f_p \approx 25.2$ MHz



HSV demodulated frame at 12 m, $f_\mathrm{c} = 350$ MHz, $B = 40$ MHz bandwidth,
$f_\mathrm{r} = 64$ MHz, $f_\mathrm{s} = 3 \times f_\mathrm{p} \approx 75.6$ MHz resampled rate, 30 frames averaged (coherently)

The challenge: cycling of TMDS 10-bit symbols

Transition Minimized Differential Signalling (TMDS)

TMDS is the 8-bit→10-bit line encoding used on DVI and HDMI cables.

- Many 8-bit bytes can be represented by one of **two alternative** 10-bit symbols.
- **DC balancing:** transmitter chooses the one that minimizes the running disparity.

  running disparity = number of 1 bits (+1) minus number of 0 bits (−1) transmitted since start of line
- Sequences of constant byte values often result in **short cycles** of TMDS symbols.

Balanced example: (cycle length 1)

$\texttt{0x10 0x10 ...} \rightarrow \overset{0}{\texttt{0111110000}} \hookleftarrow$

Unbalanced example: (cycle length 5)

$\texttt{0x0f 0x0f ...} \rightarrow \overset{-4}{\texttt{0100000101}}\ \overset{+6}{\texttt{1111111010}}\ \overset{-4}{\texttt{0100000101}}\ \overset{+6}{\texttt{1111111010}}\ \overset{-4}{\texttt{0100000101}} \hookleftarrow$

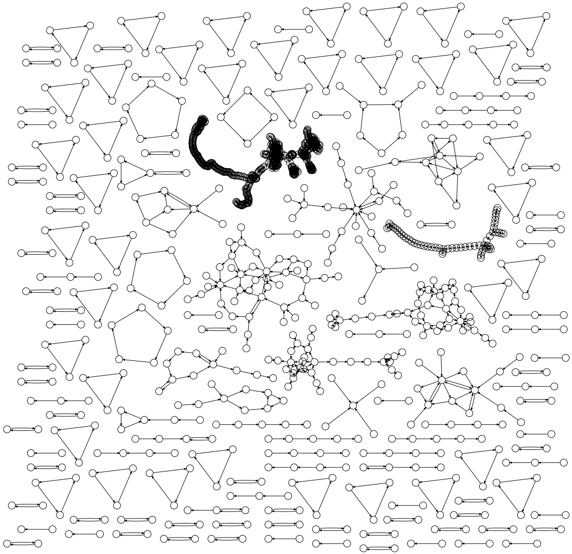
How can we **detect and remove** these cycles, to restore uniformly coloured image areas?

Idea 1

1. Use **clustering** to map received analog values (floats) for each pixel to a discrete pixel label (int) that could represent a TMDS symbol
2. Build a **directed graph of common transitions** between these cluster labels ($\approx$ TMDS symbols) and their right neighbour
3. Identify **cycles** in this graph (using *Tarjan's strongly connected components algorithm*)
4. Assign all pixels in the same **strongly connected component** the same (arbitrary) colour.

Clustering result

Graph of common transitions



Each vertex is a cluster label ($\approx$ TMDS symbol).

Each edge represents a frequent transition of cluster label from one pixel to its right neighbour.

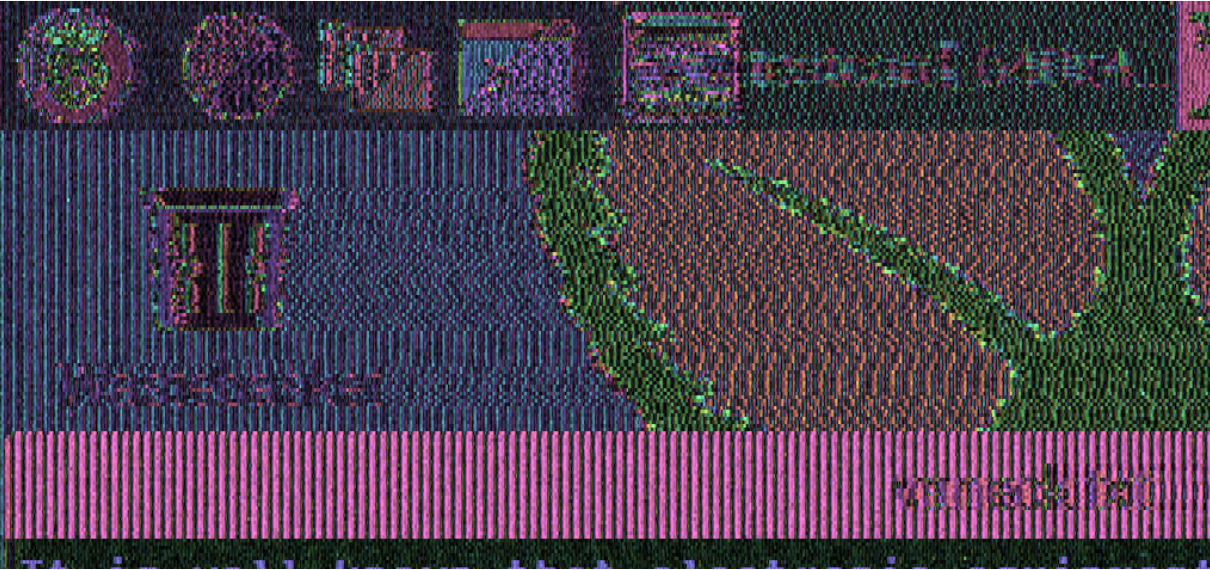
TMDS DC-balancing cycles appear as short cycles in this graph.

Strongly connected component: subset of vertices with a path between each member.

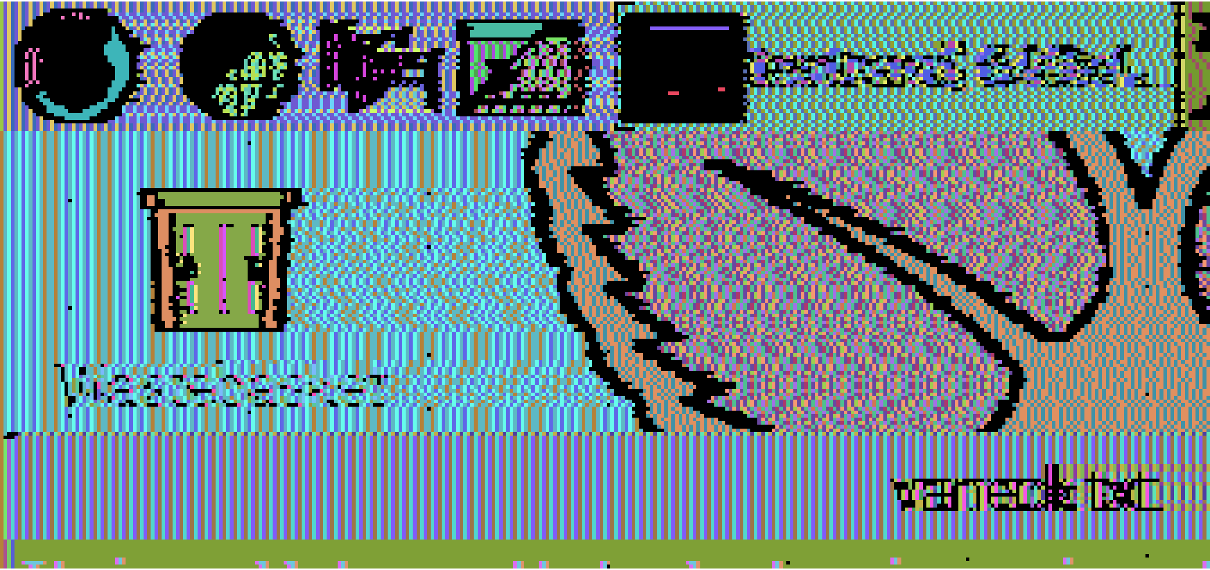
These clusters were formed based on feature data from six overlapping 40-MHz wide frequency bands: 325, 350, 375, 400, 425, and 450 MHz.

Cycle-merging result

Restored uniform background can improve readability of text

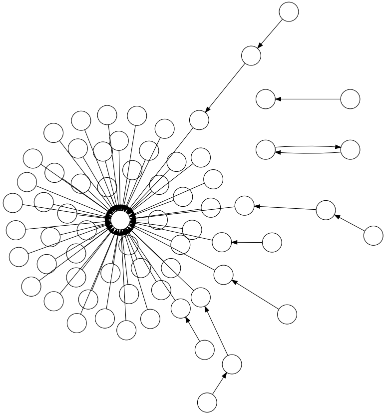


Restored uniform background can improve readability of text

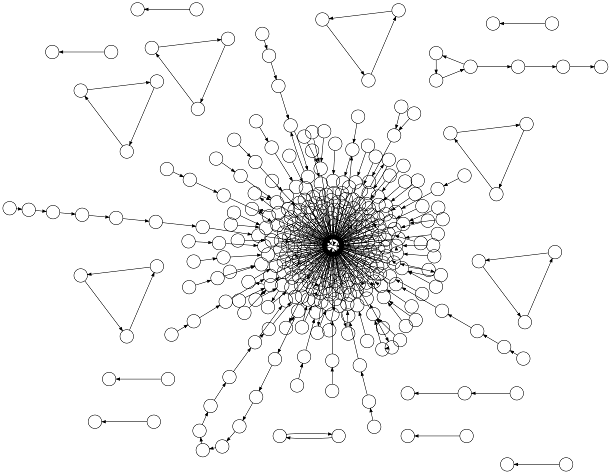


Restored uniform background can improve readability of text

Clustering attempt within single 40 MHz band: not effective



$f_c = 400$ MHz



$f_c = 425$ MHz

Idea 2

To enable effective clustering, we want a **high-dimensional feature space**.

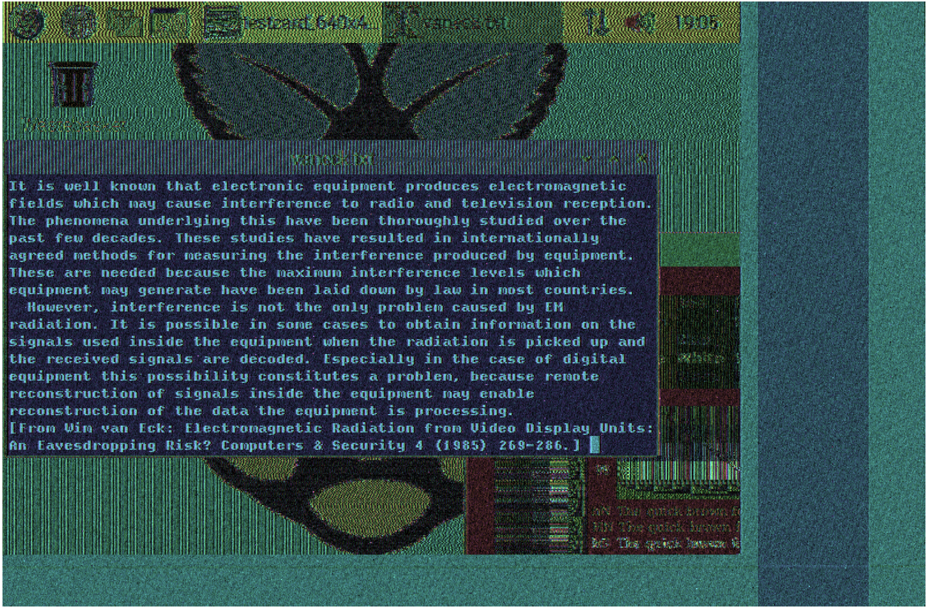
More dimensions $\Rightarrow$ better separation of clusters
$\Rightarrow$ less accidental merging of different symbols and cycles.

How many floating-point numbers can we get per pixel?

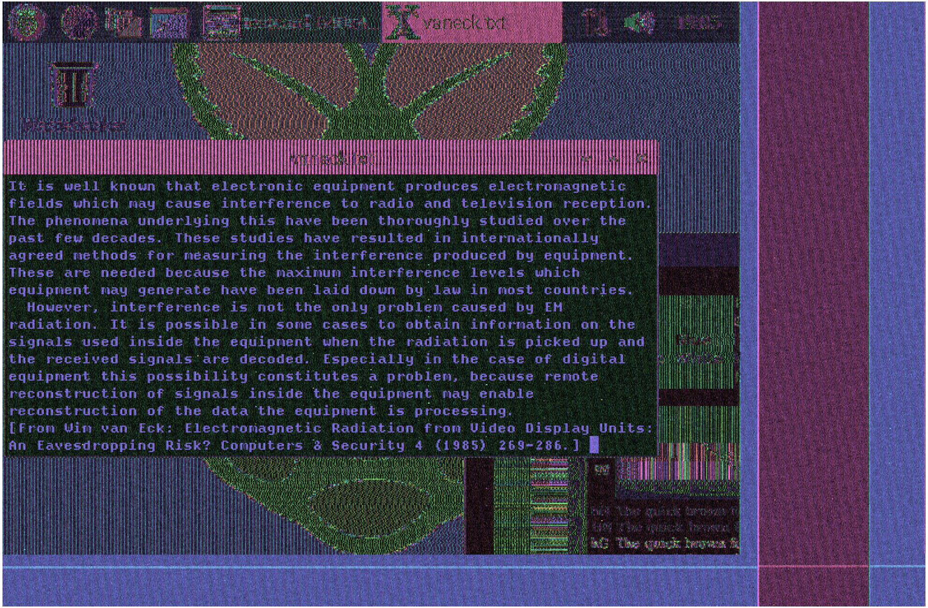
- With AM demodulation at one sample per pixel: 1 dimension
- With QAM demodulation (samples in $\mathbb{C}$): 2 dimensions
- With $f_\mathrm{r} > B > f_\mathrm{p}$: may be another $\lfloor B/f_\mathrm{p} \rfloor\times$ or $\lceil f_\mathrm{r}/f_\mathrm{p} \rceil\times$ (here $3\times$)
- With tuning into n_b different frequency bands: up to $n_\mathrm{b}\times$

In this demonstration we use QAM demodulation, 3 samples per pixel, and 6 overlapping frequency bands, resulting in a $2 \times 3 \times 6 = 18$-dimensional feature space for clustering pixels (TMDS symbols).

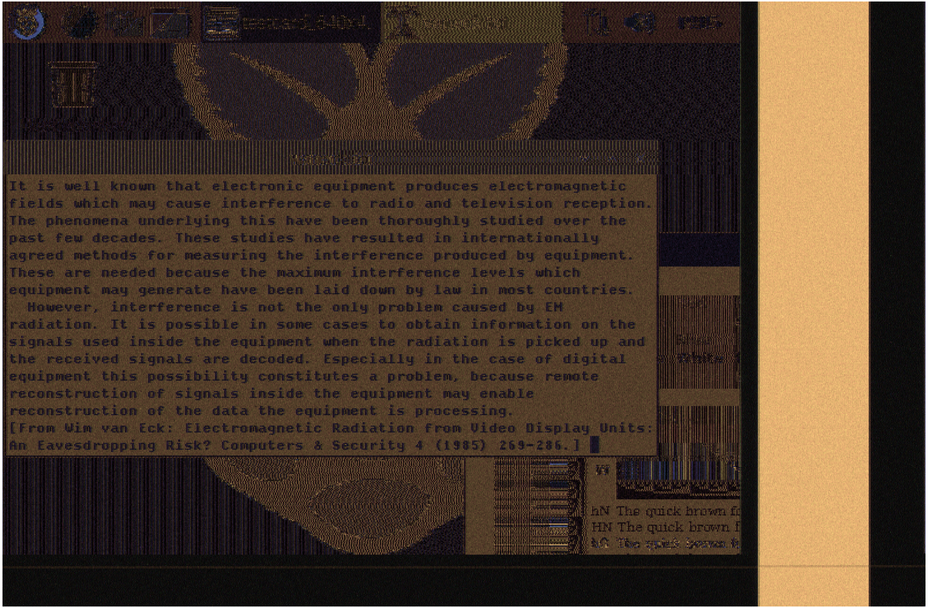
But first we need to carefully align these 6 recordings in the time and frequency domain.



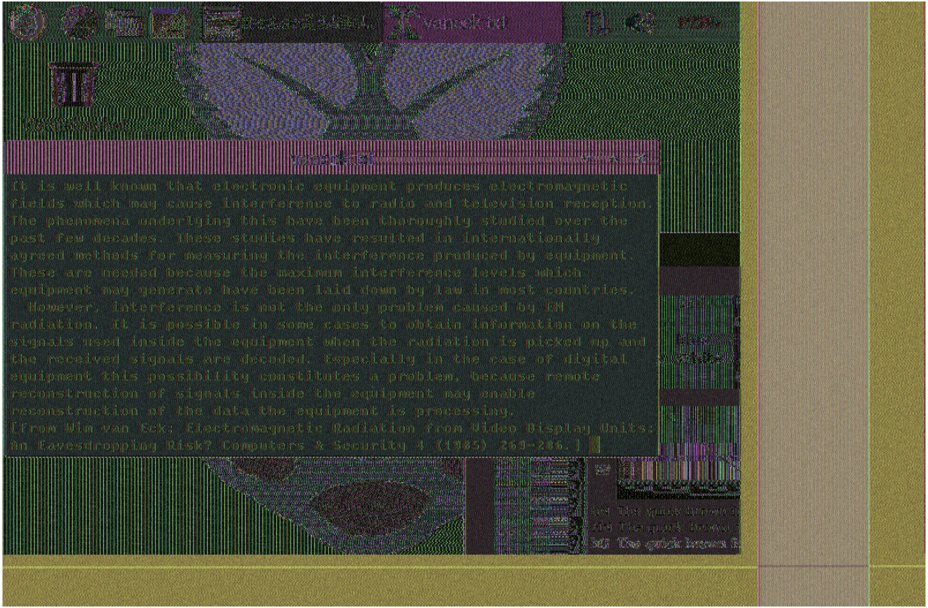
Band 1: $f_c = 325$ MHz, aligned



Band 2: $f_c = 350$ MHz, aligned



Band 3: $f_c = 375$ MHz, aligned

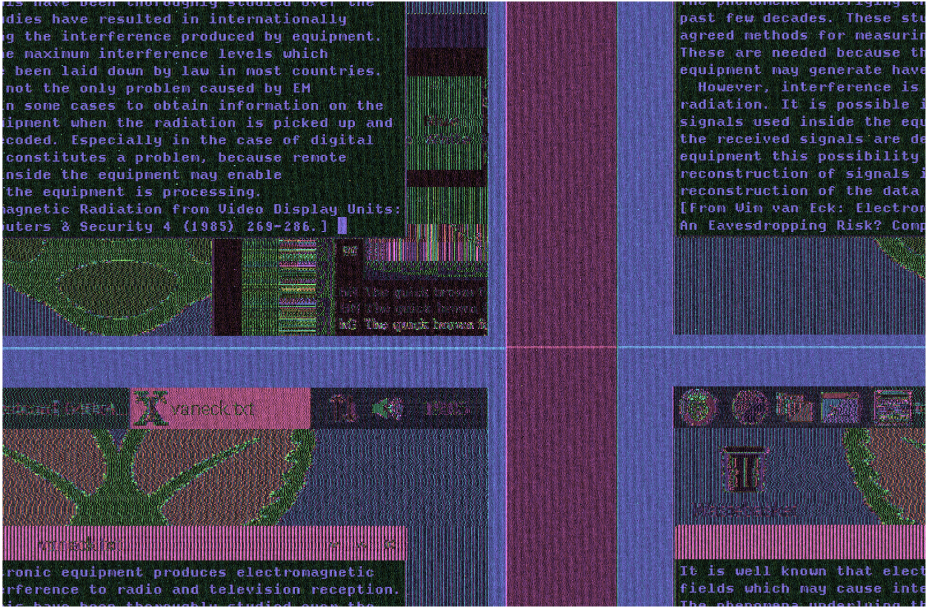


Band 4: $f_c = 400$ MHz, aligned

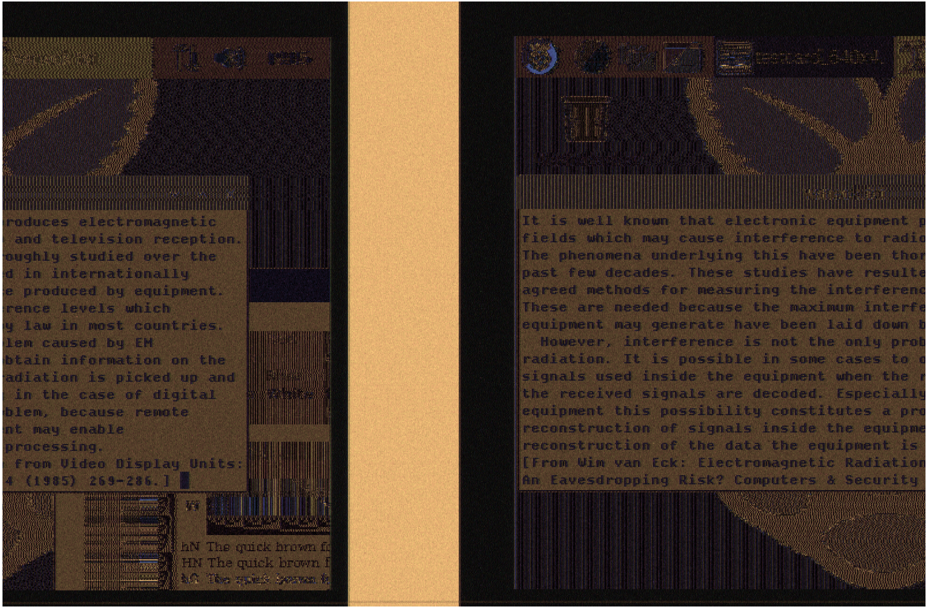
Band 5: $f_c = 425$ MHz, aligned

Band 6: $f_c = 450$ MHz, aligned

Band 1: $f_c = 325$ MHz, unaligned



Band 2: $f_c = 350$ MHz, unaligned



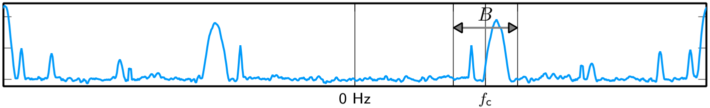
Band 3: $f_c = 375$ MHz, unaligned

In more detail . . .

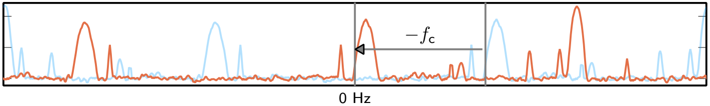
- Software-defined radio receiver
- AM vs QAM demodulation
- frequency alignment and accurate resampling (individual band)
- frequency and temporal alignment (across overlapping bands)
- clustering

Software-defined radio receiver (SDR)

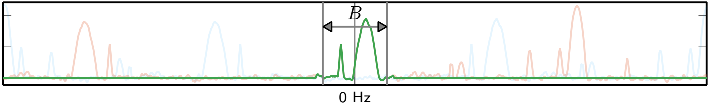
Antenna waveform (shown as Fourier spectrum) $s_{\mathsf{a}}(t)$:



Downconvert: $s_{\mathsf{d}}(t) = s_{\mathsf{a}}(t) \cdot \mathrm{e}^{-2\pi \mathrm{j} f_{\mathsf{c}} t}$

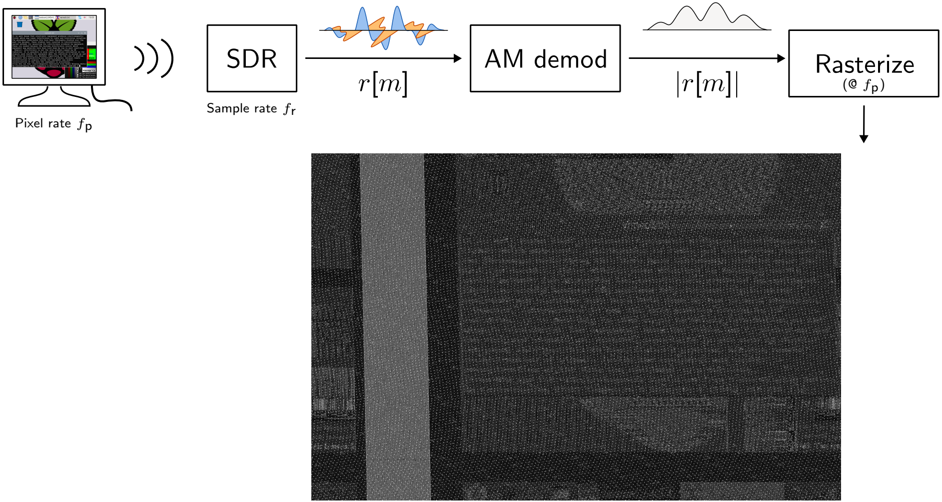


Lowpass filter: $s_{\mathsf{f}}(t) = \int s_{\mathsf{d}}(t-\tau) g(\tau) \mathrm{d}\tau$

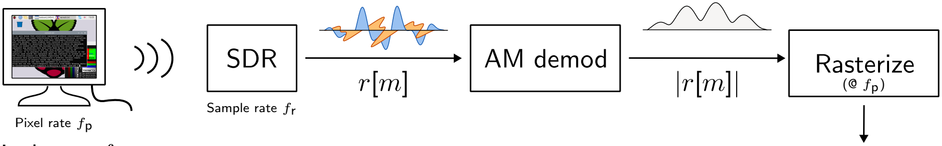


Finally, output sampled at frequency $f_{\mathsf{r}} > B$ resulting in sequence $r[m] = s_{\mathsf{f}}(m/f_{\mathsf{r}})$.

A typical TEMPEST attack



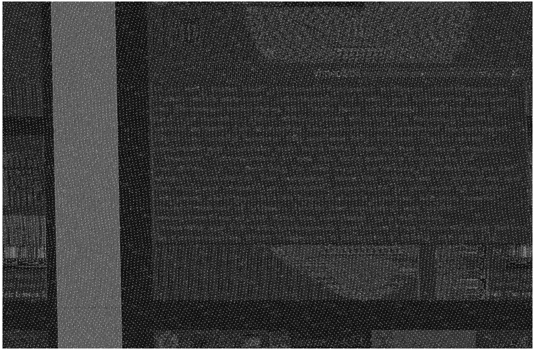
A typical TEMPEST attack



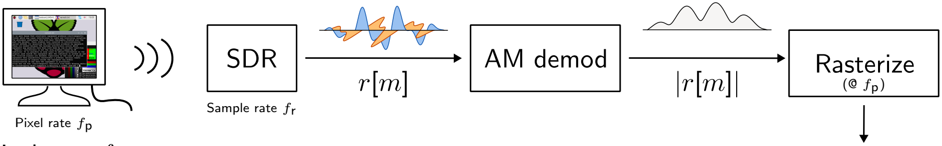
- Estimate pixel rate f_p, e.g. from the autocorrelation $R_{r,r}[d] = \sum_m r[m] \cdot r[m+d]^*$:

$$\hat{d} = \operatorname*{argmax}_{d \approx f_r / f_v} |R_{r,r}[d]|^2$$

$$f_p \approx f_r \cdot \frac{w_t h_t}{\hat{d}}$$



A typical TEMPEST attack

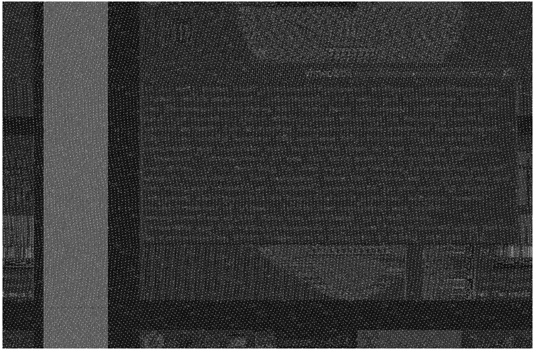


- Estimate pixel rate f_p, e.g. from the autocorrelation $R_{r,r}[d] = \sum_m r[m] \cdot r[m+d]^*$:

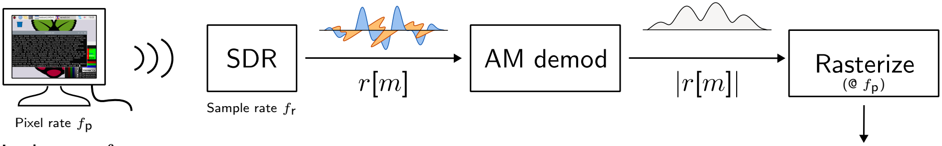
$$\hat{d} = \underset{d \approx f_r / f_v}{\operatorname{argmax}} |R_{r,r}[d]|^2$$

$$f_p \approx f_r \cdot \frac{w_t h_t}{\hat{d}}$$

- Resample to $s[n]$ at $f_s = k \cdot f_p$ for $k \in \mathbb{N}$



A typical TEMPEST attack

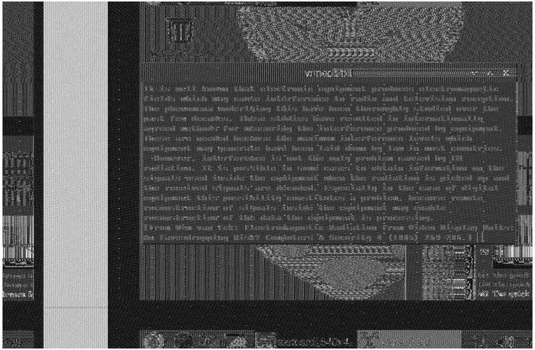


- Estimate pixel rate f_p, e.g. from the autocorrelation $R_{r,r}[d] = \sum_m r[m] \cdot r[m+d]^*$:

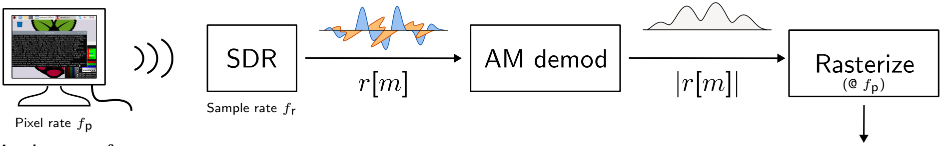
$$\hat{d} = \underset{d \approx f_\mathrm{r}/f_\mathrm{v}}{\operatorname{argmax}} |R_{r,r}[d]|^2$$

$$f_\mathrm{p} \approx f_\mathrm{r} \cdot \frac{w_\mathrm{t} h_\mathrm{t}}{\hat{d}}$$

- Resample to $s[n]$ at $f_\mathrm{s} = k \cdot f_\mathrm{p}$ for $k \in \mathbb{N}$
- Average several frames to $a[n]$



A typical TEMPEST attack

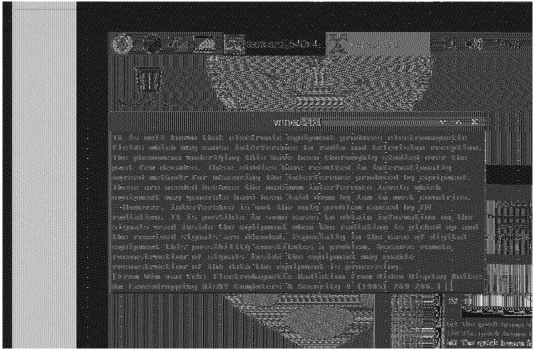


- Estimate pixel rate f_p, e.g. from the autocorrelation $R_{r,r}[d] = \sum_m r[m] \cdot r[m+d]^*$:

$$\hat{d} = \underset{d \approx f_r / f_v}{\text{argmax}} \left| R_{r,r}[d] \right|^2$$

$$f_p \approx f_r \cdot \frac{w_t h_t}{\hat{d}}$$

- Resample to $s[n]$ at $f_s = k \cdot f_p$ for $k \in \mathbb{N}$
- Average several frames to $a[n]$
- Align $a[n]$ as $k \times w_t \times h_t$ pixel raster $M_{i,j}$



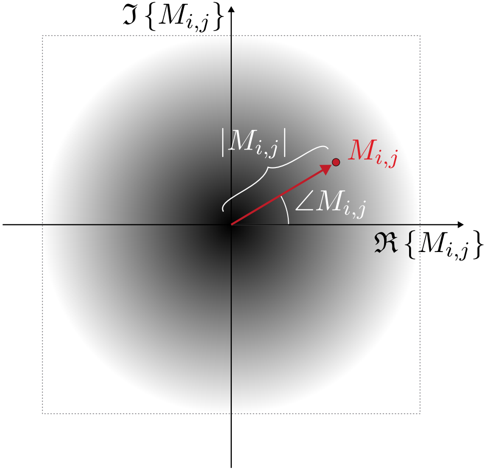
Rasterizing complex-valued signals: amplitude demodulation

Most eavesdropping demonstrations amplitude demodulate samples $M_{i,j} \in \mathbb{C}$ and visualise them as grayscale pixels.

For example, mapping 1% and 99% quantiles to black and white:

$$\text{Gray}\left(\frac{|M_{i,j}| - q_{1\%}}{q_{99\%} - q_{1\%}}\right)$$

This discards phase information $\angle M_{i,j}$.

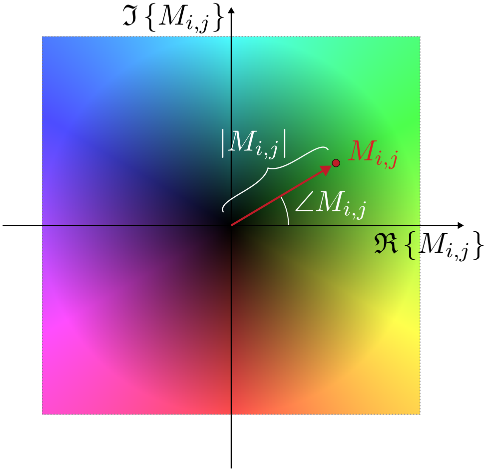


Rasterizing complex-valued signals: HSV visualisation

Using the HSV (hue, saturation, value) colour space allows us to also show the phase:

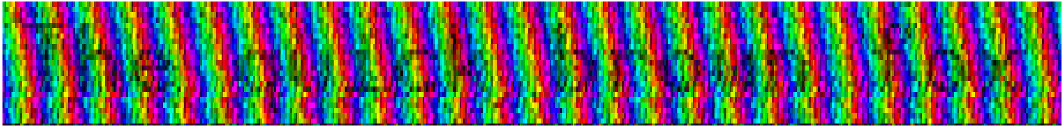
$$\text{HSV}\left(\angle M_{i,j},\ S,\ \frac{|M_{i,j}| - q_{1\%}}{q_{99\%} - q_{1\%}}\right)$$

(We leave the saturation coordinate S as a user preference.)

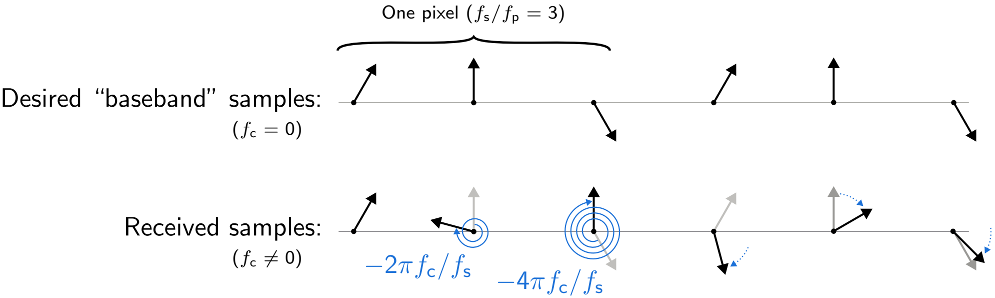


First rasterization attempt

Directly rasterizing an SDR-received signal produces a “rainbow-banding” image:

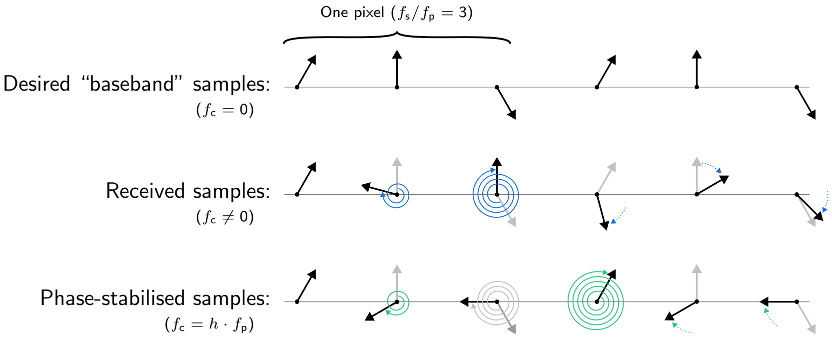


This is due to SDR downconversion from the antenna waveform $s_a(t)$ to $e^{-2\pi j f_c t} \cdot s_a(t)$.



Obtaining consistent phase angles

Shift the centre frequency to a harmonic $h \cdot f_\mathsf{p}$ of the pixel frequency:



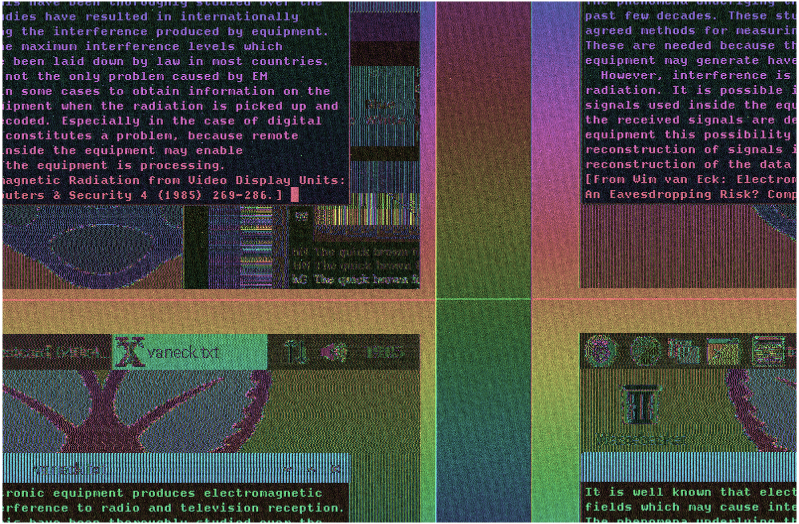
We combine frequency shifting $f_\mathsf{c} \to h \cdot f_\mathsf{p}$ with resampling to $f_\mathsf{s} = k \cdot f_\mathsf{p}$:

$$s[n] \approx \tilde{s}_\mathsf{f}\left(\frac{n+\lambda}{f_\mathsf{s}}\right) \cdot \mathsf{e}^{2\pi\mathsf{j}(f_\mathsf{c} - h f_\mathsf{p})n/f_\mathsf{s}}$$

Obtaining consistent phase angles

Some drift still remains over longer intervals.

Coherent averaging requires consistent phase across many frames, i.e. a more accurate f_p estimate.



Algorithm for accurate estimation of f_p

We improve the f_p estimate several times until convergence, by iterating over three steps:

1. Resampling and frequency-shifting $f_\mathsf{c} \to h \cdot f_\mathsf{p}$:

$$s[n] := \tilde{s}_\mathsf{f}\left(\frac{n+\lambda}{f_\mathsf{s}}\right) \cdot \mathsf{e}^{2\pi \mathsf{j}(f_\mathsf{c} - h f_\mathsf{p})n/f_\mathsf{s}}$$

2. Computing the autocorrelation:

$$R_{s,s}[qa+d] := \sum_{n=0}^{w-1} s[n] \cdot s[n+qa+d]^*$$

3. Updating the f_p estimate, with a fine-tuning term which measures phase drift between frames:

$$f_\mathsf{p} := f_\mathsf{p} \cdot \left(\frac{qa}{qa+\hat{d}} + \frac{k\angle R_{s,s}[qa+\hat{d}]}{2\pi h(qa+\hat{d})}\right)$$

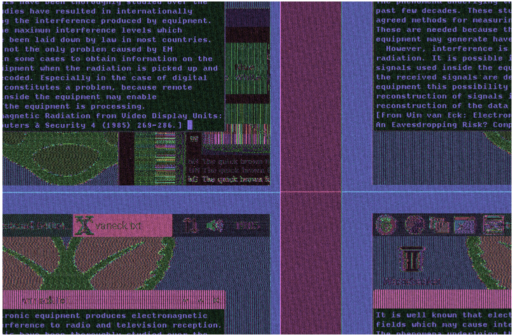
Iterations	f_p
0	25.200000000 MHz
1	25.200096064 MHz
2	25.200096764 MHz
3	25.200096794 MHz
4	25.200096793 MHz
5	25.200096788 MHz
6	25.200096788 MHz

$$q = k w_\mathsf{t} h_\mathsf{t}, \quad a = 1$$

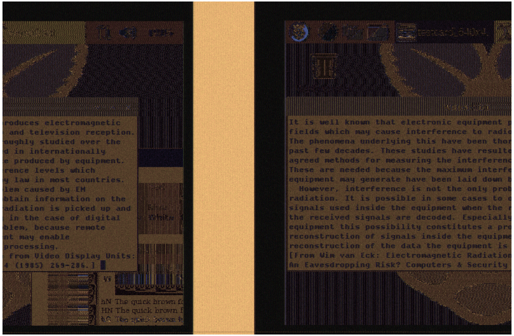
$$\hat{d} = \operatorname*{argmax}_{|d| \le d_\mathsf{max}} |R_{s,s}[qa+d]|^2$$

In later iterations, we can also search for the correlation peak $\hat{d}$ at larger multiples $a > 1$ of the frame period q.

Next: inter-band alignment of averaged frames

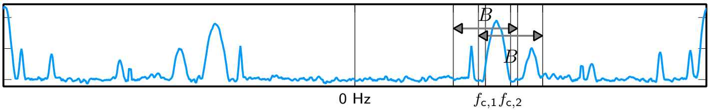


$f_c = 350$ MHz

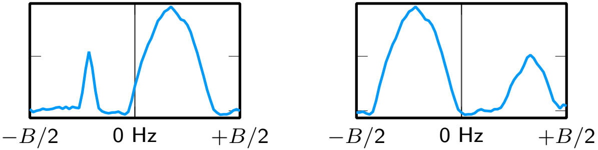


$f_c = 375$ MHz

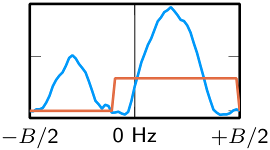
Antenna waveform (shown as Fourier spectrum) $s_a(t)$:



Downconverted, sampled and averaged waveforms $a_1[n]$ and $a_2[n]$:



$\tilde{a}_2[n] = a_2[n] \cdot e^{2\pi j n (h_2 - h_1)/k}$ (circularly) frequency shifted by $(h_2 - h_1) f_p$, then

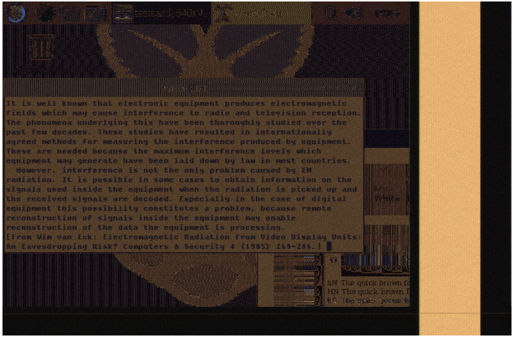


1. $A_1[v] = \text{FFT}(a_1[n])$, $\tilde{A}_2[v] = \text{FFT}(\tilde{a}_2[n])$
2. $R_{a_1\tilde{a}_2}[d] = \text{FFT}^{-1}(A_1[v] \cdot \tilde{A}_2[v]^* \cdot W_1[v])$
3. $\Delta n = \text{argmax}_d \, |R_{a_1\tilde{a}_2}[d]|^2$
4. $a_2[n] := a_2[(n + \Delta n) \bmod q]$

Alignment of averaged frames

- Align first recording $a_1[n]$ to move the blanking intervals to near the edge of the frame, via either
  - cross-correlating with manually aligned reference frame (with active pixels set to zero), or
  - identifying blanking intervals as rectangular regions of low variance
- Then align $a_2[n]$ with $a_1[n]$, $a_3[n]$ with $a_2[n]$, etc.

$f_c = 350$ MHz



$f_c = 375$ MHz

Features for clustering

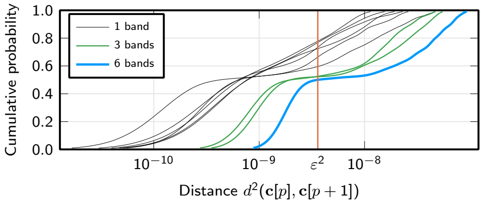
We now have averaged and aligned frames $a_1[n], a_2[n], \ldots, a_{n_\mathrm{b}}[n]$, each of length $k \times w_\mathrm{t} \times h_\mathrm{t}$.

Rearrange that data into a feature vector $\mathbf{c}[p] \in \mathbb{C}^{kn_\mathrm{b}}$ for each pixel position $0 \le p < w_\mathrm{t} \times h_\mathrm{t}$:

$$\mathbf{c}[p] = (a_1[kp+0], a_1[kp+1], \ldots, \; a_1[kp+(k-1)], \\ a_2[kp+0], a_2[kp+1], \ldots, a_{n_\mathrm{b}}[kp+(k-1)]).$$

We compare pixels using the squared L^2 norm: $d^2(\mathbf{c}[p_1], \mathbf{c}[p_2]) = (\mathbf{c}[p_1] - \mathbf{c}[p_2])^*(\mathbf{c}[p_1] - \mathbf{c}[p_2])$

For $n_\mathrm{b} = 6$ bands (or 3 non-overlapping bands) this resulted in a good bimodal distribution that can be split by a distance threshold ε^2:



Assume that TMDS symbols at pixel positions p_1 and p_2 are the same if

$$d^2(\mathbf{c}[p_1], \mathbf{c}[p_2]) < \varepsilon^2$$

Clustering

1. Initialize each pixel position with a unique cluster label $C[p] \in \mathbb{N}$.
2. Merge pixel labels such that $C[p_1] = C[p_2]$ if $d^2(\mathbf{c}[p_1], \mathbf{c}[p_2]) < \varepsilon^2$.
3. Repeat until transitive closure is reached.

Assign a different random false color to all pixels p with the same label $C[p]$.

Our label images show rare labels (< 20) as black.

Detect cycles and merge their labels

Build a graph of all remaining label values, with this set of edges:

$$E = \{(C_1, C_2) : K(C_1, C_2) \geq \eta \max_{C'} K(C_1, C')\}.$$

$K(C_1, C_2)$ counts how often label C_2 follows (immediately to the right of) label C_1. $\eta = 0.3$

Represent labels C_1 and C_2 with the same colour iff there is a path from C_1 to C_2 and one from C_2 to C_1. Can be computed efficiently with Tarjan's strongly connected components algorithm.

Conclusions

1. Coherent averaging enables high image quality and sophisticated post-processing steps
2. Automatic alignment of multiband recordings demonstrated
3. Pixel-accurate alignment possible if the frequency overlap is at least about f_p
4. Clustering into discrete pixel values is helped by availability of high-dimensional features
5. TMDS cycling can partially be countered from such cluster data (without having to fully decode TMDS symbols)

Limitations and potential future work:

1. We used a static target image (unchanged for 6×0.5 s recordings)
2. Currently random assignment of display colours to merged cluster labels, but a time series of output images would benefit from consistent colour assignment
3. Processing time: tens of seconds $\Rightarrow$ faster algorithms, GPU acceleration