Multi-frame Denoising and Color Emanation Analysis for TempestSDR Image Reconstruction

Terézia Gurová, Richard Ostertág*

Department of Computer Science, Faculty of Mathematics, Physics and Informatics, Comenius University, Bratislava, Slovakia

Abstract

Monitors and video cables unintentionally emit an electromagnetic field during image transmission, which can be picked up by a software-defined radio and used by the TempestSDR tool to reconstruct the original image. However, the resulting image has two fundamental drawbacks: it is heavily degraded by noise, and it loses color information. This paper addresses both problems. In the first part, we exploit the fact that when a static screen is observed, the useful content remains constant across consecutive captured frames, while the noise behaves randomly. We propose a pipeline that automatically captures a series of consecutive frames of the same screen, mutually aligns them, and statistically combines them into a single output image with substantially lower noise. The pipeline was experimentally evaluated on a series of captured frames, comparing two alignment algorithms and three statistical combination methods. The resulting image is significantly more readable than any individual captured frame. In the second part, we analyze the loss of color information. We identified three potential sources of radiation in the test setup: the TMDS signal on the HDMI cable, the analog VGA signal, and the monitor's internal LVDS interface. For each, we build a model that maps the original pixel value to the expected captured intensity. By quantitatively comparing these models against measured data from captured gradients, we have shown that the dominant source of radiation is the TMDS encoding of the HDMI signal.

Keywords

electromagnetic side-channel attack, TempestSDR, noise reduction, image registration

1. Introduction

Electronic devices unintentionally emit electromagnetic fields during operation, and this radiation carries information about the data being processed. This radiation, known as compromising emanations, can be intercepted at a distance and used to recover processed data without physical access to the device. This phenomenon has been studied since the mid-20th century by the American NSA under the classified TEMPEST program [1], which examined a wide range of equipment from computers and monitors to printers and keyboards.

The first public demonstration of such an attack on a computer monitor was conducted by Wim van Eck in 1985, who showed that an image displayed on a CRT screen can be reconstructed from its electromagnetic emanations using commonly available electronics [2]. The question of whether the attack extends to later display technologies was answered systematically by Markus Kuhn, whose 2003 technical report [3] extended the attack to LCD monitors and digital video interfaces and analyzed in detail the mechanisms by which compromising emanations arise in modern display chains. A further important step was the 2014 work of Martin Marinov [4], who released the open-source tool TempestSDR. The tool reconstructs an image from a signal captured by a software-defined radio (SDR), making the attack accessible to anyone with a commodity SDR receiver and antenna.

However, the image obtained in this way has two fundamental drawbacks. The first is strong noise, arising from interference from surrounding electronic equipment and the limited strength of the useful signal, which often makes the reconstructed image difficult to read. The second is the loss of color information. Although the video signal carries three color components in parallel, the antenna captures only one signal of the resulting field, and the output is therefore monochromatic. Moreover, the

ITAT'26: Information Technologies – Applications and Theory, September 25–29, 2026, Biele Karpaty, Slovakia

*Corresponding author.

gurova16@uniba.sk (T. Gurová); richard.ostertag@fmph.uniba.sk (R. Ostertág)

0000-0002-6560-1515 (R. Ostertág)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-09

CEUR-WS.org/Vol-4323/paper40.pdf

relationship between the original pixel color and its captured intensity is not intuitive. Due to the way signals are encoded in modern digital interfaces, a pixel with higher intensity can produce a weaker emission than a pixel with lower intensity.

In this paper, we address both problems. In the first part, we exploit the fact that when a static scene is observed, the image itself remains constant between consecutive captures, while the noise behaves randomly. By combining multiple consecutive frames, it is therefore possible to statistically suppress noise. In the second part, we analyze the loss of color information itself. We describe which part of the transmission chain contributes the most to the emanation. We also discuss the implied relationship between the original color and the captured intensity.

2. Experimental setup

Our setup follows the configuration of [5], which we briefly describe here for completeness. The target is a laptop with HDMI output connected via an HDMI-to-VGA converter to a BenQ LCD monitor at a resolution of 1024×768 pixels and a refresh rate of 60 Hz, with a nominal pixel clock of $\approx 65\,\mathrm{MHz}$. The signal therefore goes through several physically distinct stages on its path from the GPU to the display: TMDS encoding on the HDMI cable, decoding and digital-to-analog conversion inside the converter, three analog RGB voltages on the VGA cable, analog-to-digital conversion at the monitor input, and finally a serial differential LVDS link to the LCD panel's row drivers. Each of these stages is a potential source of EM emanation, which becomes central in Section 4.

Emanations are captured with a National Instruments USRP-2944R SDR receiver tuned to the second harmonic of the pixel clock (130 MHz) with sampling at 20 MS/s. The receiving antenna is positioned a short distance from the monitor and cables; the exact distance is determined by the experimental goal. For the noise-reduction experiments of Section 3, we capture the signal from a moderately larger distance so that noise dominates and frame-stacking has an observable effect. For the color analysis of Section 4, we capture from close range so that the underlying signal is as clean as possible. The image reconstruction is performed by TempestSDR [4], running on Linux because its UHD plugin for USRP receivers is officially supported only on Linux and macOS.

3. Multi-frame noise reduction

The image reconstructed by TempestSDR is significantly degraded by noise. However, when observing a static scene, this noise varies across captured frames, while the image itself remains practically unchanged. This property can be exploited by capturing a sufficient number of frames of the same scene and combining them appropriately. In this way, random noise can be statistically suppressed, yielding a higher-quality image.

3.1. Pipeline

We extended the TempestSDR GUI with a function that automatically stores 50 consecutive reconstructed frames as PNG files. The number 50 balances two effects. The relative noise level after averaging N independent frames scales proportionally to $1/\sqrt{N}$ [6], so the gain of each additional frame quickly diminishes, while the whole capture must be completed in a short time, so that the assumption of a static scene is valid.

Even with a stationary monitor and antenna, the frames within a single series differ not only in noise. The static screen content appears at slightly different positions, typically shifted by a few pixels. The cause is TempestSDR's per-frame synchronization recovery, which estimates the start of every line and frame from periodic patterns in the noisy captured signal. Averaging without compensating for these shifts would blur the result. We therefore align each frame to the first frame of its series before combining.

Three families of image registration methods are commonly used [7]. Feature-based methods such as SIFT [8] or ORB [9] are not suitable for TempestSDR data. Low contrast and high noise cause feature detectors to respond primarily to noise, resulting in keypoints that are unstable across frames. We compare two representatives of the remaining families: phase correlation [10, 11], a frequency-domain method that estimates translations from the linear phase shift in the cross-power spectrum, and the Enhanced Correlation Coefficient (ECC) algorithm [12], which iteratively maximizes the mean-subtracted normalized correlation between the reference and a shifted target. ECC is more expensive per call, but it tolerates lower signal-to-noise ratios and global brightness drift better than phase correlation does, both of which are present in our captures. We use the translational variant of ECC with tolerance 10^{-6} and a cap of 1,000 iterations.

After alignment, we combine 50 frames pixel-wise by one of three estimators: arithmetic mean, per-pixel median, and iterative sigma-clipping [13] with $k = 3$ and at most five iterations.

3.2. Results

We cannot directly compare the reconstructed image with the displayed "source" reference: the two differ in geometry and intensity mapping (for details, see section 4). Instead, we used three indirect, no-reference metrics.

First, to evaluate alignment, we use the *mean inter-frame standard deviation*, computed for each pixel across the aligned frames, and averaged spatially. In perfect alignment, this reflects only the captured noise; in poor alignment, misregistered edges contribute additional variability, and the value increases. Table 1 reports the mean inter-frame standard deviation across the 87 series. Phase correlation *worsens* the result relative to no alignment: at the very low signal-to-noise ratio of TempestSDR captures, its correlation peak flattens out. ECC reduces the inter-frame standard deviation below the unaligned baseline.

Alignment strategy	**Mean inter-frame ↓ standard deviation**
No alignment	8.79
Phase correlation	10.18
ECC	7.69

Table 1
Mean inter-frame standard deviation across 87 captured series, by alignment strategy.

Second, to evaluate the combined output, we use two metrics. *Tenengrad sharpness* [14] is the mean squared gradient magnitude of the image; higher values indicate sharper edges. The *border noise* is the standard deviation in the synchronization margins, which contain no image content and, therefore, estimate residual noise. Both metrics are indirect. Tenengrad reacts to duplicated edges from misalignment as well as true edges, and border noise rises when alignment shifts the previously static margins along with the image content. We complement the numbers with visual inspection.

Table 2 reports Tenengrad and border noise for every combination of alignment strategy and combination method. The median yields the highest Tenengrad but also the highest border noise. Mean and sigma-clipping are indistinguishable on both metrics. We adopt sigma-clipping for its robustness against outlier frames. Figure 1 shows the overall effect on a representative text region.

4. Color emanation analysis

The Multi-frame combination addresses the noise problem but leaves the second drawback of TempestSDR output untouched. The TempestSDR output is monochromatic. The antenna captures a single scalar projection of the EM field, with contributions from all three color channels and all stages of the transmission chain. The reconstructed image is therefore a function $I = f(R, G, B)$ that collapses

Alignment	Method	Tenengrad ↑	Border noise ↓
No alignment	mean	822.7	19.0
	median	930.1	19.6
	sigma-clip	835.8	19.1
Phase correlation	mean	1,080.9	24.5
	median	1,385.4	27.0
	sigma-clip	1,113.4	24.8
ECC	mean	1,078.3	24.2
	median	1,294.9	25.9
	sigma-clip	1,103.2	24.5

Table 2
Tenengrad sharpness and border noise for every combination of alignment strategy and combination method, averaged over 87 captured series. Higher Tenengrad indicates stronger edges; lower border noise indicates less residual noise.

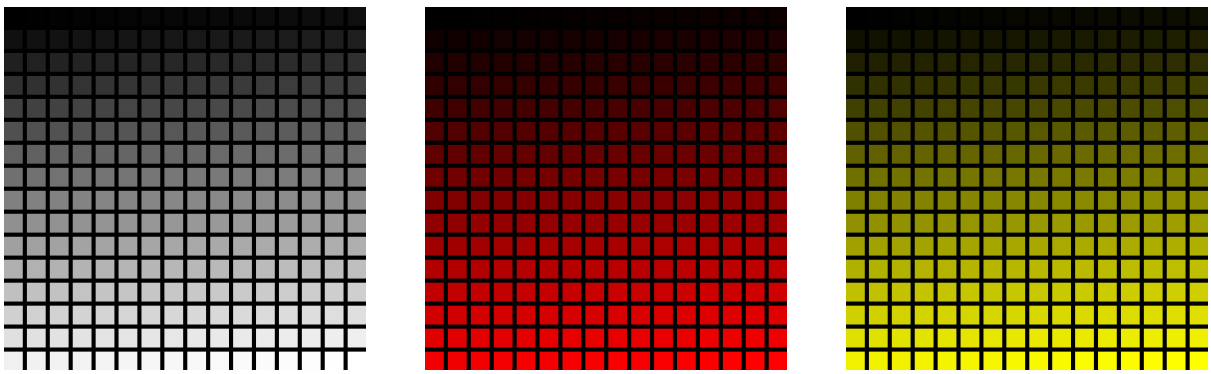


(a) captured frame (b) sigma-clipping

(c) captured frame, contrast enhanced (d) sigma-clipping, contrast enhanced

Figure 1: Effect of multi-frame combination on a representative text region, shown at original contrast (top) and with the same level of contrast enhancement for better visibility (bottom). On the left side is one captured frame. On the right is the sigma-clipping ($k = 3$) of 50 ECC-aligned frames. The combined output is substantially more readable than any single frame.

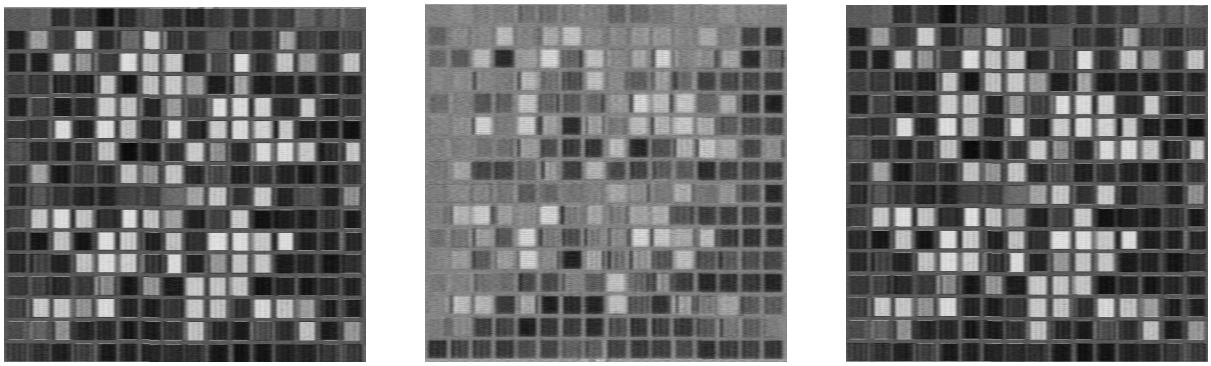
three input dimensions into one. As a first empirical look at the properties of f, we displayed a 16×16 grid of squares whose color values ramp from $(0, 0, 0)$ to a target color in steps of one. We tested all six basic and complementary primary gradients (to red, green, blue, cyan, magenta, yellow) plus white. An example of the gradients is shown in Fig. 2. The corresponding captured images are shown in Fig. 3.



(a) gray gradient (b) red gradient (c) yellow gradient

Figure 2: Examples of test gradients. (R, G, B): (a) $= (i, i, i)$; (b) $= (i, 0, 0)$; (c) $= (i, i, 0)$, for $i = 0, \ldots, 255$.

The result is surprising. The captured grid is *not* a monotonic gradient. Visually similar input colors can produce very different output intensities, and colors at opposite ends of the numeric range can produce similar output. Analogical non-monotonic patterns appear in all other gradients we have tested.



(a) gray gradient (b) red gradient (c) yellow gradient

Figure 3: Examples of captured gradients reconstructed by TempestSDR.

4.1. Candidate radiation sources

The transmission chain in our setup consists of three signal segments with substantially different physical characteristics, each of which is a plausible source of image-correlated EM emission.

HDMI / TMDS. The 8-bit color output values of the GPU are encoded into 10-bit symbols by Transition-Minimized Differential Signaling (TMDS) before being serialized and transmitted over the HDMI cable [15]. The bit rate is ten times the pixel clock, i.e. 650 Mbit/s in our setup. TMDS is *stateful*. The encoder maintains a running disparity counter that tracks the imbalance between ones and zeros sent so far on the line and adjusts encoding decisions to keep the line DC-balanced [16]. The same 8-bit input can therefore be encoded into different 10-bit symbols depending on what was sent before in the same line.

Analog VGA. Inside the HDMI-to-VGA converter, the decoded RGB values are fed to three digital-to-analog converters that drive three analog voltages in the 0.0 V – 0.7 V range [17]. The VGA cable then carries these voltages directly to the monitor. The amplitude of EM emanation from each conductor is approximately proportional to dV/dt, so the expected captured-intensity-versus-input function is approximately linear.

Internal LVDS. After the VGA signal is re-digitized at the monitor input, the pixel data is forwarded to the LCD row drivers via an internal LVDS link. The standard 24-bit single-link FPD-Link variant serializes seven bits per pixel clock onto each of four differential pairs [18]. Unlike TMDS, LVDS is stateless (the bit pattern on the line depends only on the current pixel). However, the fixed bit-by-bit assignment of color and synchronization signals to the four pairs means that similar pixel values produce structurally different bit patterns, again without a general monotonic relation to emission intensity.

4.2. TMDS encoding in detail

Because TMDS will be the dominant source in our setup, we describe its encoding in more detail.

TMDS encoding proceeds in two steps [15, 16]. The input is an 8-bit byte $D = (D_0, D_1, \ldots, D_7)$ representing one color channel of one pixel; the output is a 10-bit symbol $q^{\text{out}} = (q_0, q_1, \ldots, q_9)$ serialized onto the differential pair.

Step 1: transition minimization. The encoder forms an intermediate 9-bit word q^m:

$$q_0^m = D_0, \qquad q_i^m = q_{i-1}^m \oplus D_i \quad \text{for } i = 1, \ldots, 7, \tag{1}$$

where $\oplus$ is either XOR or XNOR. The choice depends on the Hamming weight of D: XNOR is used if D has more than four ones, or exactly four ones with $D_0 = 0$; XOR otherwise. The rule is chosen so that $q_{0..7}^m$ has at most four bit transitions, regardless of D. A flag bit $q_8^m \in \{0, 1\}$ tells the decoder which operation was used.

Step 2: DC balancing. The encoder maintains an integer counter cnt, the *running disparity*, representing the current imbalance between ones and zeros transmitted so far on the line. Given q^m and the current cnt, the encoder chooses whether to invert the lower eight bits of q^m before transmission. Inversion is signaled by an extra output bit q_9^{out}, and cnt is updated based on how many ones and zeros the symbol actually contributed. The detailed update rules are given in [15]. The practical effect is that the same input D can, depending on cnt, produce one of two different 10-bit symbols. One of these is the complement of the lower eight bits of the other.

Why this is non-monotonic. The strength of EM emanation generated by transmitting a symbol scales approximately with the number of *transitions* ($0 \to 1$ or $1 \to 0$) it contains. Each transition is a steep voltage edge on the line, generating high-frequency spectral content. Crucially, the transition count of a TMDS symbol depends on the bit pattern of the input, not on its numerical value. Table 3 shows this for several inputs encoded with $cnt = 0$. Inputs at the extremes of the numeric range ($D = 0$ and $D = 255$) produce symbols with almost no transitions, because their input bit patterns themselves have no transitions; inputs with alternating patterns ($D = (85)_{10} = (01010101)_2$ and $D = (170)_{10} = (10101010)_2$) produce symbols with the maximum transition count. The numerical distance between $D = 0$ and $D = 128$ is large, yet they produce the same transition count and thus radiate similarly. This is the fundamental mechanism behind the non-monotonic pattern in Fig. 3.

Table 3
TMDS symbols (i.e. q^{out}) for representative input values, computed with running disparity $cnt = 0$. The transition count, not the numeric value of D, predicts emanation strength.

D	Binary D	TMDS symbol	Transitions
0	00000000	0000000010	2
85	01010101	1100110010	5
128	10000000	0000000110	2
170	10101010	1100110001	4
255	11111111	0000000001	1

4.3. Spectral content and the 130 MHz harmonic

TMDS runs at 650 Mbit/s, but TempestSDR demodulates near 130 MHz a factor of five lower. This is possible because the TMDS bit stream is not a single sinusoid at 650 MHz, but it is a quasi-periodic serialized stream that transmits one 10-bit symbol per pixel clock. Its spectrum, therefore, consists of discrete lines at every multiple of the pixel-clock frequency $f_p = 65\,\text{MHz}$,

$$f_n = n \cdot f_p, \qquad n = 1, 2, 3, \ldots, \tag{2}$$

with line amplitudes that depend on the bit patterns of the transmitted symbols and therefore carry image information [3]. Any of these lines is a viable demodulation target; we choose $n = 2$ (130 MHz) because it yields the cleanest reconstruction in our setup.

4.4. Simulating the TMDS emanation

To turn the qualitative argument into a quantitative prediction, we simulate the full TMDS bit stream together with the SDR demodulation pipeline, with no free parameters fitted to the captured data.

The simulation is parameterized by the pixel clock $f_p = 65\,\text{MHz}$, the TMDS bit rate $f_b = 10 f_p = 650\,\text{Mbit/s}$, the SDR carrier frequency $f_c = 130\,\text{MHz}$, and the receiver's half-bandwidth $B = 10\,\text{MHz}$ (from the 20 MS/s sample rate). It is convenient to work in units of cycles per bit, in which the normalized carrier is $F_c = f_c/f_b = 0.2$ and the normalized cutoff is $F_B = B/f_b \approx 0.0154$.

For each video line, we encode the sequence of 8-bit pixel values through the TMDS encoder of Sec. 4.2, carrying the running disparity from one pixel to the next as on the real link. The resulting 10-bit

symbols are concatenated into a bit stream, which we map to a bipolar line signal $s[n] \in \{-1, +1\}$ representing the differential voltage on the wire. We then mix this stream to baseband at the carrier frequency, apply a low-pass filter matching the receiver's bandwidth, and take the magnitude of the resulting complex envelope. Finally, we averaged the envelope within each 10-bit window to obtain a single intensity value per pixel. The complete signal-processing core fits in a dozen lines of Python (Listing 1).

Listing 1: Core of the TMDS emanation simulation. The encoder `tmds_encode` follows the algorithm of Sec. 4.2; `F_CARRIER` and `F_CUTOFF` are the normalized carrier and bandwidth in cycles per bit.

```
def tmds_channel_power(pixel_values):
   # 1) TMDS-encode the whole line; carry disparity
   cnt, bits = 0, []
   for val in pixel_values:
      sym, cnt = tmds_encode(int(val), cnt)
      bits.extend(sym)
   bits = np.array(bits, dtype=float)

   # 2) Bipolar line signal
   sig = 2.0 * bits - 1.0

   # 3) Mix to baseband at f_c = 130 MHz
   n = np.arange(len(sig))
   baseband = sig * np.exp(-2j*np.pi*F_CARRIER*n)

   # 4) Low-pass filter matching SDR bandwidth
   win = max(1, int(round(1.0 / F_CUTOFF)))
   filtered = np.convolve(baseband, np.ones(win)/win, mode='same')

   # 5) Envelope, averaged over 10 bits = 1 pixel
   return np.abs(filtered).reshape(-1, 10).mean(axis=1)
```

To predict the captured intensity of one grid square, we build a video line consisting of the appropriate sequence of color values (i.e. 40 pixels of constant value followed by 10 pixels of black for the gap, repeated across all 16 columns of the test grid) and feed it through the simulation for each color channel separately. For example, for the yellow gradient $(R, G, B) = (i, i, 0)$, both the red and green channels contribute, so the returned envelopes for the two channels are summed to predict the total intensity. The mean of the predicted envelope over each square's 40 pixels yields a single intensity per square.

For comparison, we also implemented two alternative models. The LVDS model assembles the four 7-bit FPD-Link patterns from the RGB triple and computes the spectral power of each at the second harmonic of the pixel clock, summing the contributions of the three channels. The VGA model is the trivial monotonic linear fit $I(i) = ai + b$. The next section compares the three models against the captured data.

4.5. Quantitative results

To compare each model with the captured data. We extract a single representative intensity from each grid square. For this purpose, we used the mean over a centered 30×30-pixel region, chosen to avoid edge artifacts from imperfect alignment. We compare the predictions of each model against the 256 measured intensities using the Pearson correlation r and the coefficient of determination R^2, which produces the results shown in Table 4. Because Pearson is invariant to linear shift and rescaling, R^2 here is the fraction of variance in the measured data explained by the shape of the model's curve, after the optimal linear rescaling.

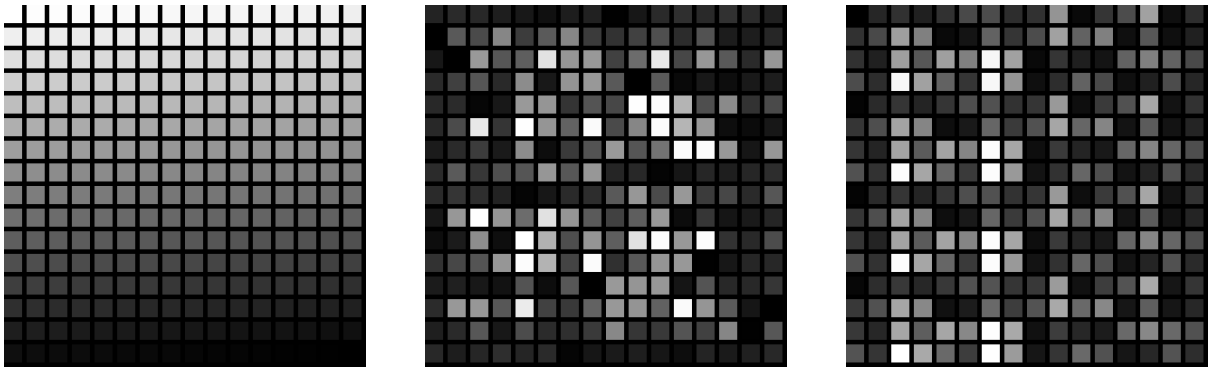
The TMDS bit-stream model alone explains approximately 78 % of the variance in the measured intensities, while the LVDS and VGA models explain only 2 % and essentially zero, respectively. The TMDS model has no parameters fitted to the data. Everything is derived from the published TMDS specification [15, 16] and the receiver bandwidth quoted in Section 2. For this reason, we consider this

Table 4
Quantitative agreement of the three emanation-source models with the captured yellow gradient (256 samples). None of the models has free parameters fitted to the data beyond the linear shift and scale that Pearson correlation absorbs.

Model	Pearson r	R^2
VGA (linear in i)	+0.05	0.00
LVDS bit stream	+0.14	0.02
TMDS bit stream	+0.89	0.78

to be a strong quantitative confirmation that the dominant emanation source in our test setup is the TMDS-encoded HDMI signal. The visual agreement between the simulated TMDS grid in Fig. 4(b) and the captured grid in Fig. 3(b) is also striking. The locations of bright and dark squares match, including the lateral structure within each row.

The remaining 22 % of unexplained variance has several plausible contributors that the idealized model does not capture. The most significant is the residual EM noise. The receiver picks up the desired TMDS signal along with broadband background emissions and its own thermal noise, whereas our simulation does not. The second is the state-dependence of TMDS. Because the encoder's running disparity depends on the entire history of the line, the encoding of a constant-color square, in fact, varies slightly along its 40-pixel width, an effect averaged out but not removed by our per-square mean. The smaller contributors include additive emanation from the analog VGA and LVDS segments (consistent with the small positive correlations of those models in Table 4) and small positioning errors when locating the per-square crop in the captured image.



(a) VGA (b) TMDS (c) LVDS

Figure 4: Simulated gradients

4.6. Implication for color reconstruction

A practical consequence is that the mapping $f : (R, G, B) \to I$ is not only non-monotonic but also non-injective. Many RGB triples map to similar captured intensities. Deterministic inversion of f from a single monochromatic capture is therefore principally impossible. Recovering color information from TempestSDR output requires additional knowledge, typically statistical priors about which colors and color combinations are plausible in a given scene. The quantitative TMDS model we have validated here provides the physical model upon which future learning-based approaches can be built.

5. Conclusion

We presented two complementary improvements to TempestSDR-based image reconstruction. A multi-frame statistical pipeline exploits the static character of typical screen content. By capturing 50

consecutive frames, aligning them using the Enhanced Correlation Coefficient algorithm, and combining them via sigma-clipping, we have reduced the mean inter-frame variability from 8.79 to 7.69 across 87 captured series, whereas the more standard phase-correlation alignment paradoxically increases it to 10.18. This approach does not use training data and can serve as a standalone denoiser or as a preprocessing step before learning-based reconstruction, as in [5].

A model-based analysis quantitatively identifies the TMDS-encoded HDMI signal as the dominant source of image-correlated emanations in our test setup. The TMDS bit-stream model without free parameters explains approximately 78 % of the variance in the measured intensities of captured color gradients, whereas alternative LVDS and analog VGA models explain only a few percent. This pinpoints the physical reason why the captured intensity-to-color mapping is non-monotonic and non-injective. It is the bit-transition count of the TMDS 10-bit symbols, not the original pixel value, that drives EM emanation in this setup. However, the TMDS model provides a physical basis for future statistical or learning approaches to partial color recovery.

Acknowledgments

This publication is the result of support under the Operational Program Integrated Infrastructure for the project: Advancing University Capacity and Competence in Research, Development and Innovation (ACCORD, ITMS2014+:313021X329), co-financed by the European Regional Development Fund.

Declaration on Generative AI

During the preparation of this work, the authors used Claude, ChatGPT, and Grammarly for grammar and spelling checks, stylistic editing, and assistance with the drafting of the source code. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] National Security Agency, TEMPEST: A Signal Problem, Cryptologic Spectrum 2 (1972) 26–30. Declassified document.

[2] W. van Eck, Electromagnetic radiation from video display units: An eavesdropping risk?, Computers & Security 4 (1985).

[3] M. G. Kuhn, Compromising emanations: eavesdropping risks of computer displays, Technical Report UCAM-CL-TR-577, University of Cambridge, Computer Laboratory, 2003. URL: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-577.pdf. doi:10.48456/tr-577.

[4] M. Marinov, Remote Video Eavesdropping Using a Software-Defined Radio Platform, Master's thesis, University of Cambridge, 2014.

[5] F. Tuch, R. Ostertág, Reconstruction of the image captured by TempestSDR, in: Proceedings of ITAT 2025: Information Technologies – Applications and Theory, volume 4092 of *CEUR Workshop Proceedings*, 2025.

[6] A. V. Oppenheim, A. S. Willsky, Signals and Systems, Prentice Hall, 1999.

[7] B. Zitová, J. Flusser, Image registration methods: A survey, Image and Vision Computing 21 (2003) 977–1000. doi:10.1016/S0262-8856(03)00137-9.

[8] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110.

[9] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An efficient alternative to SIFT or SURF, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 2564–2571.

[10] C. D. Kuglin, D. C. Hines, The phase correlation image alignment method, in: Proceedings of the IEEE International Conference on Cybernetics and Society, 1975, pp. 163–165.

[11] B. S. Reddy, B. N. Chatterji, An FFT-based technique for translation, rotation, and scale-invariant image registration, IEEE Transactions on Image Processing 5 (1996) 1266–1271. doi:10.1109/83.506761.
[12] G. D. Evangelidis, E. Z. Psarakis, Parametric image alignment using enhanced correlation coefficient maximization, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 1858–1865. Hal-00864385.
[13] M. P. Maples, D. E. Reichart, N. C. Konz, T. A. Berger, A. S. Trotter, J. R. Martin, D. A. Dutton, M. L. Paggen, R. E. Joyner, C. P. Salemi, Robust Chauvenet outlier rejection, The Astrophysical Journal Supplement Series 238 (2018) 2. doi:10.3847/1538-4365/aad23d.
[14] S. Pertuz, D. Puig, M. A. Garcia, Analysis of focus measure operators for shape-from-focus, Pattern Recognition 46 (2013) 1415–1432. doi:10.1016/j.patcog.2012.11.011.
[15] HDMI Licensing, LLC, High-Definition Multimedia Interface Specification, Version 1.3a, Technical Report, HDMI Licensing, LLC, 2006.
[16] Digital Display Working Group, Digital Visual Interface (DVI), Specification, Digital Display Working Group, 1999.
[17] C. Poynton, Digital Video and HD: Algorithms and Interfaces, Morgan Kaufmann, 2012.
[18] Texas Instruments, FPD-Link Design and Implementation Guide, Texas Instruments, 1998.