Benefits of Coherent Demodulation for Eavesdropping on HDMI Emissions

Dimitrije Erdeljan, Markus G. Kuhn

Department of Computer Science and Technology
University of Cambridge, UK
{de298,mgk25}@cl.cam.ac.uk

Abstract — **Demonstrations of electromagnetic eavesdropping attacks on digital video-interface cables usually first amplitude-demodulate the received signal, then resample the result into pixel-aligned raster images, and finally average consecutive video frames obtained this way. This non-coherent process discards useful phase information. Software-defined radio receivers allow us to coherently demodulate and average such signals, using a linear process that preserves phase information. We describe techniques for performing phase-coherent resampling and periodic averaging of compromising emanations from HDMI video cables and demonstrate how the preserved phase information enables better discrimination of on-screen colours.**

Keywords — **TEMPEST, compromising emanations, eavesdropping, displays, software radio, information leakage, side-channel attacks**

I. INTRODUCTION

Reconstructing readable text from unintentional electromagnetic emissions of video-interface cables is particularly practical if the target signal is periodic at the frame-refresh rate, which provides much redundancy. Early video eavesdropping demonstrations, both on cathode-ray tube displays [1] and flat-panel displays with Digital Visual Interface (DVI) [2] used amplitude-demodulating analog receivers. More recent demonstrations using digital software-defined radio (SDR) receivers (e.g., [3], [4]) continued to perform non-coherent amplitude demodulation, but also explored quadrature amplitude demodulation [5, Ch. 4] and frequency demodulation [6].

In this paper, we provide a quick tutorial on how to process SDR-received emissions in a phase-coherent manner that enables complex-valued periodic averaging before any demodulation takes place. We then show how this generally improves contrast and readability and demonstrate how some supposedly more secure combinations of background and foreground colours, chosen to provide low contrast after AM demodulation, are much easier to distinguish if phase information is preserved this way.

II. SOFTWARE-DEFINED RADIO RECEIVER

A *software-defined radio* (SDR) receiver is a general-purpose data-acquisition device that captures and digitises a chosen part of the radio-frequency spectrum. The resulting discrete sequence of samples $r[m]$ can be streamed to a computer, where demodulation and other signal-processing steps can then be implemented entirely in software. The main user-configurable parameters of an SDR receiver are the centre frequency f_c of its reception band $\left[f_c - \frac{B}{2}, f_c + \frac{B}{2}\right]$ and its sampling rate f_r, which must be greater than the desired bandwidth B (e.g., $f_r \geq 1.25 \times B$).

Let $s_0(t)$ be the analog input to the SDR, the antenna voltage at time t. The receiver first *downconverts* that, shifting the spectrum in the frequency domain by $-f_c$, which in the time-domain corresponds to multiplying with a complex phasor with frequency $-f_c$:

$$s_d(t) = s_0(t) \cdot e^{-2\pi j f_c t}. \tag{1}$$

This frequency shift centres the reception band around 0 Hz, and, since the spectrum is no longer symmetric around 0 Hz, results in a complex-valued baseband signal $s_d(t)$. This is then lowpass filtered with cut-off-frequency $\frac{B}{2}$, to eliminate signals outside the reception band and avoid aliasing:

$$s_f(t) = \int s_d(t - \tau)g(\tau)d\tau. \tag{2}$$

Here $g(\tau)$ is the impulse response of the filter. Finally, sampling the result produces the output discrete-time sequence

$$r[m] = s_f(m/f_r). \tag{3}$$

III. VIDEO EAVESDROPPING BASICS

If $s_m(t)$ is the signal unintentionally emitted by the targeted display hardware, for example the sum of filtered versions of the twisted-pair waveforms found in a High-Definition Multimedia Interface (HDMI) cable, then we receive $s_0(t) = s_m(t) + n(t)$, where the noise term $n(t)$ shall cover both thermal noise and uncorrelated interference from other sources. If the displayed image is static, we assume here that $s_m(t)$ will be periodic, with *frame period* f_v^{-1}.

The downconverted and filtered signal $s_f(t)$ will similarly contain a component that is periodic, up to multiplication with $e^{-2\pi j f_c/f_v}$ between frames (i.e., a phase rotation). We, therefore, can estimate the frame period f_v^{-1} by first estimating the autocorrelation

$$R_{rr}[d] = \mathbb{E}\left[r[m]\,r[m + d]^*\right] \tag{4}$$

and then searching for a magnitude peak, considering a range $[f_{v,\perp}, f_{v,\top}]$ of typical frame rates (e.g. $f_v(1 \pm \frac{1}{2h_t})$ or 60 Hz $\pm$ 0.1%):

$$f_v \approx f_r \left/ \underset{f_{v,\perp} \leq f_r/d \leq f_{v,\top}}{\operatorname{argmax}} |R_{rr}[d]|^2 \right. . \tag{5}$$

Fig. 1. In this HSV rasterization of the resampled SDR data, prior to approximate frequency correction, the horizontal period length of the rainbow banding is $f_s / |f_c - hf_p|$ samples, and therefore indicates how far away from 0 Hz the nearest pixel-frequency harmonic hf_p is after initial downconversion by the SDR.

Assuming the video mode of the target display has a total image width of w_t pixels and height of h_t lines, including blanking periods, its pixel rate will be

$$f_p = w_t h_t f_v. \tag{6}$$

To rasterize the received signal, we first resample (interpolate) $r[m]$ into a new discrete sequence

$$s[n] \approx s_f\left(\frac{n+\lambda}{f_s}\right) \tag{7}$$

with sampling rate $f_s = k f_p$, such that each displayed pixel corresponds to $k \in \mathbb{N}$ samples, for example $k = \lceil f_r / f_p \rceil$. An offset $\lambda \in [0, k)$ can optionally be added to adjust alignment between samples and pixels, and larger offsets can be added to align $s[0]$ with the first pixel of a frame.

We can then reshape $k w_t h_t$ consecutive samples of $s[n]$ into a complex-valued $h_t \times k w_t$ matrix M. A basic demonstration of amplitude-demodulating video eavesdropping then takes all the absolute values $|M_{i,j}|$ of that matrix, and linearly maps, for example, their 1% and 99% quantiles to black and white, respectively, outputting the result as a $k h_t \times k w_t$ pixel large greyscale image, repeating (or vertically stretching) each row k times to obtain an image where now each $k \times k$ square corresponds to a displayed pixel.

We can improve the resulting signal-to-noise ratio by *periodically averaging* the $|M_{i,j}|$ values obtained from N consecutive sequences of $k w_t h_t$ samples from $s[m]$, as long as we know the target's f_v with a relative accuracy better than $1/(N w_t h_t)$.

IV. COHERENT DEMODULATION

The HSV (hue, saturation, value) colour space offers a simple way to also incorporate the phase information into the rasterized image, by mapping the angle $\angle M_{i,j}$ to the circular *hue* coordinate, while setting the *value* coordinate like the previously generated greyscale value:

$$\text{HSV}\left(\angle M_{i,j}, S, \frac{|M_{i,j}| - q_{1\%}}{q_{99\%} - q_{1\%}}\right) \tag{8}$$

The remaining *saturation* coordinate S can be left as a user preference, with 0 resulting in the previous greyscale image.

Such images will initially show a rapid "rainbow banding" effect (Figure 1), because we have yet to compensate for the multiplication by a complex phasor $e^{-2\pi j f_c t}$ during downconversion in (1). If all pixels in the target image (including blanking intervals) were encoded as the same bit sequence, we would want to see the same hues repeated every k samples of $s[n]$. In this case, $s_m(t)$ would be a periodic function with
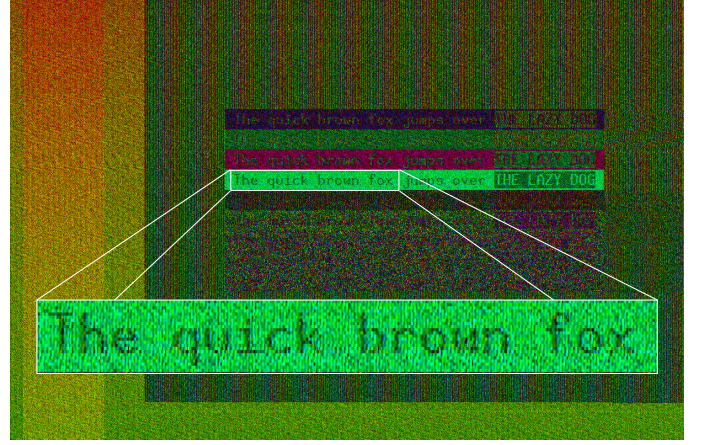

Fig. 2. The first frequency adjustment using equation (9), still based on the first f_p estimate from the autocorrelation peak according to (5) and (6), increases the period length of the rainbow banding in this single HSV raster frame to more than one frame duration, meaning that $|f_c - hf_p| < f_v$. The rainbow banding now appears vertically, and can be best seen in the homogeneous signal of the horizontal blanking interval (at the left of the frame here).

period length f_p^{-1} and its Fourier spectrum would only contain energy at integer multiples of f_p. Therefore, we can obtain consistent phase angles (and hues) for each pixel by further shifting the frequency spectrum of $s_f(t)$ such that one of the harmonics (integer multiples) of f_p ends up at 0 Hz. For this we pick the h-th harmonic of the pixel rate, that is hf_p, which with $h = \lfloor f_c / f_p + 0.5 \rfloor$ is already closest to 0 Hz in $r[m]$.

Starting from our previous estimate of f_p, obtained from (5) and (6), we can now resample $r[m]$ again with $f_s = k f_p$, while also applying a frequency shift of $f_c - hf_p$, such that

$$s[n] \approx s_f\left(\frac{n+\lambda}{f_s}\right) \cdot e^{2\pi(f_c - hf_p)n/f_s}. \tag{9}$$

When we rasterize that $s[n]$ sequence, the previous "rainbow banding" effect has been reduced to a much slower hue drift, in the order of one rotation per frame (Figure 2), resulting in a more readable image.

While this correction is accurate enough to stabilize the phase of $s[n]$ over a short period (e.g. a fraction of a frame), it still remains too large for frame-periodic averaging of the complex values in $s[n]$. To fix this, we need an even closer estimate of f_p. From the autocorrelation peak location

$$\hat{d} = \underset{f_{v,\perp} \leq f_r/d \leq f_{v,\top}}{\operatorname{argmax}} |R_{ss}[d]|^2. \tag{10}$$

on $s[n]$, we can update our estimate of f_p as

$$f_p := f_p \cdot \left(\frac{k w_t h_t}{\hat{d}} + \frac{k \angle R_{ss}[\hat{d}]}{2\pi h \hat{d}}\right). \tag{11}$$

This also gives us an updated sampling rate $f_s := k f_p$, which we can then apply in the next iteration back in (9), (10) and (11), repeating these updates until $R_{ss}[\hat{d}]$ is real valued (e.g. $\left|\angle R_{ss}[\hat{d}]\right| < 1°$). The first term in (11) updates f_p to move the location $\hat{d}$ of the magnitude peak to the exact length $k w_t h_t$ of one frame in $s[n]$. The second term, which is much
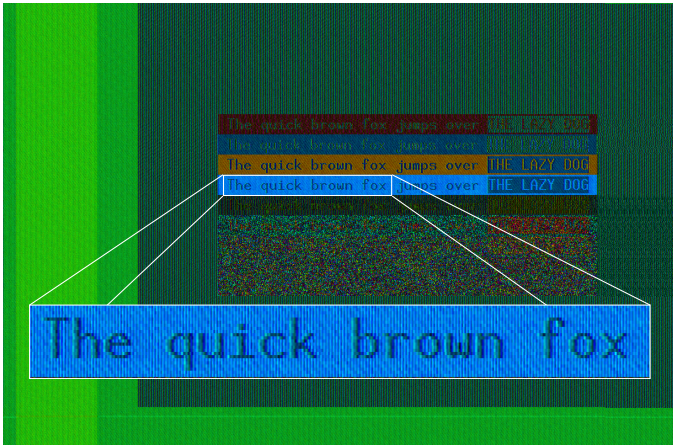
Fig. 3. After the frequency correction through equations (9–11) has converged (3 iterations here), the rainbow banding disappears completely, and the resulting coherently resampled signal now enables periodic averaging of the complex values $M_{i,j}$. This HSV raster image shows the result of averaging the complex values of 30 frames.

Fig. 4. Target and antenna, in the atrium of a normal office building

Fig. 5. The test image, which is shown centred on the target display.

smaller, then fine-tunes f_{p} such that the angle of the frame-rate peak $R_{ss}[\hat{d}]$ of the autocorrelation of $s[n]$ goes to zero. Together, these terms converge such that we sample $s_{\mathrm{f}}(t)$ at an integer multiple of the actual pixel rate. In our experiments, this approximation converges after just a couple of iterations. (The width of the search interval $[f_{\mathrm{v},\perp}, f_{\mathrm{v},\top}]$ can be much reduced in later iterations, around an updated f_{v} estimate.)

The resulting phase-stabilized sequence $s[n]$ can then be periodically averaged as complex samples, before any demodulation or rasterization (Figure 3).

Complex-valued averaging of N frames requires that f_{p} has been approximated with a relative error well below $1/(Nhkw_{\mathrm{t}}h_{\mathrm{t}})$, such that the remaining phase rotation across the averaging interval remains well below one rotation.

In practice, we have observed that the frequencies of the crystal oscillators in eavesdropping targets and SDR receivers perform random walks that cause the remaining phase rotation to change by up to $30°$ within a dozen frames. Therefore, to enable longer-term averaging, a tracking loop has to be implemented to compensate for continuous changes in the pixel rate f_{p}. See O'Connell [5, Ch. 4] for a detailed description of a phase-locked loop implementation that does this.

If the eavesdropper successfully tracks and compensates the phase rotation of the downconverted SDR samples as described here, by using the updated f_{p} estimate for both phase unrotation and resampling, this will automatically also lead to a stable alignment of the pixel grid, and therefore prevents the eavesdropped image from moving sideways, a synchronization problem that otherwise would have to be dealt with separately in amplitude-demodulated non-coherent methods. This is made possible by the fact that in a typical SDR receiver, the sampling clock and the local oscillator used for frequency downconversion are derived from the same time base, and therefore only a single drifting frequency ratio has to be tracked, between the time bases of the SDR receiver and the eavesdropping target.

V. Practical Demonstration

We now demonstrate how coherent demodulation can help an eavesdropper to improve the contrast and readability of received text.

The eavesdropping target for our experiments was a Raspberry Pi computer (model B+ Rev 1.2), connected via a 1.8 m HDMI cable and an HDMI-to-DVI adapter to a DVI monitor (Dell 1704FPT), using the video mode 640×480 @ 60 fps ($w_{\mathrm{t}} = 800$, $h_{\mathrm{t}} = 525$, $f_{\mathrm{p}} \approx 25.20$ MHz). We recorded the emissions using a UHF log-periodic antenna (Schwarzbeck VULSP 9111B) at 5 m distance from the target, connected to a receiver (R&S FSV7 in I/Q-analyzer mode). We recorded the antenna signal at a tuning frequency of $f_{\mathrm{c}} = 425$ MHz with sampling rate $f_{\mathrm{r}} = 64$ MHz (bandwidth $B = 40$ MHz). A photo of the setup (antenna and target) can be seen in Figure 4.

Figure 5 shows the 450×216-pixel test image displayed by the target (summarised in Table 1), which shows the same text in several different combinations of foreground and background colours, including some chosen to maximize (line 4) or minimize (line 5) the contrast after amplitude demodulation, as well as attempts to hinder eavesdropping by randomising less-significant bits (lines 6–9), as proposed in [2].

Line 3 shows a colour pair where the bit patterns are complements of each other, also aimed as a countermeasure against amplitude-demodulating eavesdroppers. The transferred data bytes `0x10` and `0xee` appear on the HDMI cable as `0111110000` and `1000001111`, respectively, after an 8/10-bit encoding step known as Transition-Minimized Differential Signaling (TMDS). Since these bit sequences are identical up to a half-pixel phase shift, we expect the resulting electromagnetic emissions from data lanes to have identical amplitudes for both colours, and to differ only in phase.

Table 1. 24-bit RGB colours used in the test image

	description	foreground			background		
1	black and white	00	00	00	ff	ff	ff
2	maximum bit transition contrast	00	00	00	aa	aa	aa
3	complementary TMDS encoding	10	10	10	ee	ee	ee
4	balanced TMDS encoding, max contrast	10	10	10	55	55	55
5	minimum contrast	ff	00	00	00	ff	00
6	low nibble random	fx	fx	fx	0x	0x	0x
7	text in msb (two for green), rest random	–			–		
8	text in two msb green bit, rest random	–			–		
9	text in msb green bit, rest random	–			–		

(a)   (b)
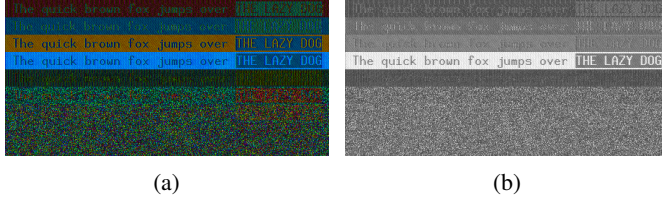
Fig. 6. (a) cut-out of the same HSV image of 30 averaged frames as in Fig. 3; (b) the same 30 frames averaged after non-coherent AM demodulation.
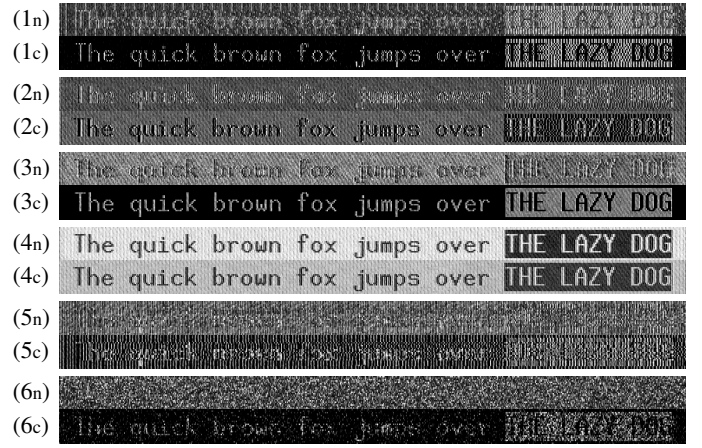

Fig. 7. Pairwise comparison of the first six lines from the test image. The top line in each pair shows the result of averaging 30 frames after *non-coherent* AM demodulation (as in Fig. 6); the bottom line shows instead the result of projecting the complex value obtained after *coherent* averaging of pixels from 30 frames, resulting in various degrees of improved contrast.
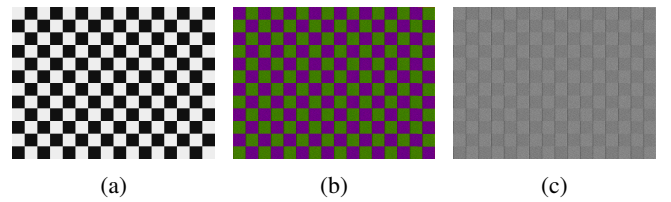

(a)   (b)   (c)

Fig. 8. (a) checkerboard test image of RGB colours #101010 and #eeeeee, which result in complementary TMDS bit patterns 0111110000 and 1000001111, (b) after coherent HSV rasterization, and (c) after AM demodulation.

We used a 0.5 s long recording (≈ 30 frames) to precisely determine the pixel rate f_p and coherently resampled the received signal at $f_\mathrm{s} = 3f_\mathrm{p}$, as described in Section IV. We then averaged the complex samples of the pixels in 30 consecutive frames. The result can be seen in HSV colours in Figure 6a. For comparison, the rasterized image in Figure 6b was produced using the same recording and estimated f_p, but with the signal amplitude-demodulated before periodic averaging, as described in Section III.

To illustrate more clearly the readability benefits that coherent demodulation can bring, we compare in Figure 7 greyscale rasterizations obtained after either non-coherent or coherent averaging, for each of the first six lines from the test image in Figure 5. The upper image in each pair is the respective line amplitude demodulated as $|M_{i,j}|$. However, this discards sign information. In the lower image, we instead project $M_{i,j}$ onto a line at angle φ in the complex plane, that is, we map $\Re\left\{e^{j\varphi} M_{i,j}\right\}$ to pixel brightness. The brightness in each image extract was again normalized such that the 1% quantile value appears black and the 99% one white.

We adjusted φ separately for each line in the test image to maximize contrast. For each line, this resulted in the text becoming easier to read in the coherently demodulated image. The largest difference is, as expected, in line 3, which uses colours chosen such that they are difficult to distinguish from amplitude alone. As line 6 shows, the protective effect of randomising the four least-significant bits of each image byte, which very effectively prevents eavesdropping using an amplitude-demodulating receiver, is also partially overcome by coherent demodulation and resulted in identifiable text.

We used a second test image (Figure 8a) to further compare the performance of amplitude demodulation versus amplitude-and-phase based coherent colour detection methods. It consists of 40×40 squares alternating between the colours used in line 3 of the previous test image, namely #101010 and

#eeeeee. We again processed a 0.5 s long recording to obtain averaged complex-valued pixels $M_{i,j}$, as in the previous section. The resulting image, visualised as HSV colours, is shown in Figure 8b.

We compare this with amplitude-demodulated samples $|M_{i,j}|$, shown as a greyscale image in Figure 8c. Note that, unlike in the previous demonstration, here we coherently averaged complex-valued samples before amplitude demodulation.

After aligning the displayed and rasterized images, we split samples $|M_{i,j}|$ into two groups, based on the colour of the corresponding displayed pixel, discarding samples close to the boundary lines of the checkerboard field. The distributions of
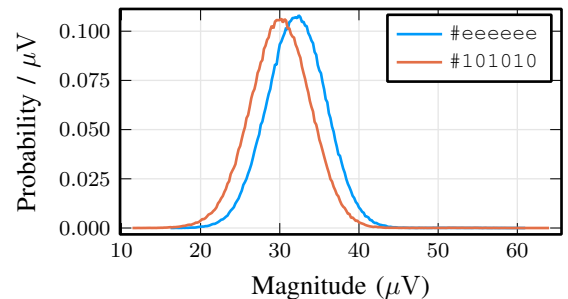

Fig. 9. Amplitude distributions of the two colours from Fig. 8c
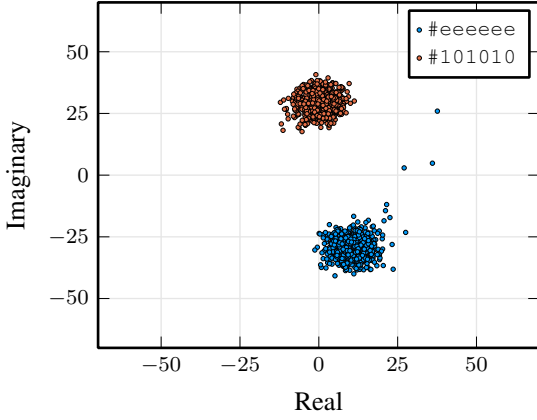
Fig. 10. Complex-valued distributions of the two colours with complementary TMDS encodings from Fig. 8b
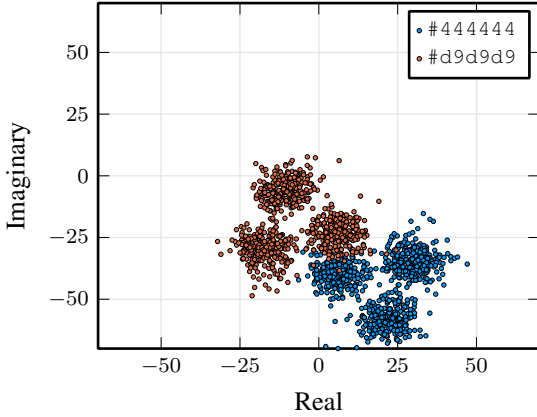

Fig. 11. Complex-valued distributions of two colours with non-complementary, but still DC-balanced, TMDS encodings, each resulting in $k = 3$ clusters, dependent on the position of the sample within the pixel.

Fig. 12. The amplitude distributions of the samples for the two colours from Fig. 11 also show significant overlap.

amplitudes in these two groups overlap significantly, as Figure 9 shows. Distinguishing these two colours from amplitudes of individual pixels alone is, therefore, hardly possible.

Using the complex values instead allows us to accurately separate them. Their respective distributions, shown in Figure 10 as a scatter plot of 0.2% of the samples of one averaged frame, do not overlap.

The two cluster centres in Figure 10 are not quite negatives of each other (i.e., π out of phase). Two possible reasons come to mind. Firstly, the HDMI clock signal, which is transmitted on a fourth wire pair, can add a constant bias. While data signals transferred for the two colours would be negatives of each other, the clock signal remains the same. Secondly, the symmetry may also be broken by imbalances in line drivers in the HDMI source. These would result in differing impulse responses due to rising and falling edges. Either effect would also explain the different mean amplitudes seen in Figure 9, since for exactly opposite signals, their amplitude distributions would be identical.

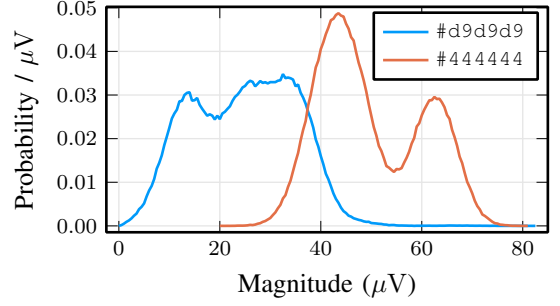Next, we repeat the experiment with another checkerboard

image, with RGB colour squares #d9d9d9 and #444444. Unlike the previous colour pair, which we had chosen to result in low contrast if amplitude demodulated, this time we picked the colours at random from among all greyscale colours whose byte values are, after TMDS encoding, always represented using the same 10-bit sequence (i.e., their encoding is DC balanced and therefore independent of the DC-balancing TMDS encoder's state, see [5, Appendix A]).

The resulting complex-valued and amplitude distributions are shown in Figures 11 and 12, respectively. The TMDS encodings 1011100010 and 0100111100 are more varied, with multiple bit transitions. Since we have $k = 3$ samples per pixel, each colour can result in three clusters, one for each of the three positions that a sample can have within a pixel.

VI. DIMENSIONALITY REDUCTION

Since the k complex-valued samples per pixel can be differently distributed, we can also consider their combined distribution over a $2k$-dimensional space: two complex coordinates for each of the k sample positions within a pixel. We can then use dimensionality-reduction techniques, such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA), to project this $2k$-dimensional signal for each pixel onto either a 1-dimensional greyscale axis, or even a 2- or 3-dimensional colour space, with projection axes chosen to maximize signal variance.

Since in our experiments $f_\mathrm{s} = 3f_\mathrm{p}$, we obtain for each pixel three complex numbers, or six real numbers after we split them into real and imaginary parts:

$$\mathbf{x}_{i,j} = (\Re\{M_{i,3j}\}, \Im\{M_{i,3j}\}, \ldots, \Im\{M_{i,3j+2}\}) \in \mathbb{R}^6.$$

We now perform Principal Component Analysis, that is we estimate a $6{\times}6$ covariance matrix $\mathbf{B}$ to describe the distribution of these 6-dimensional vectors $\mathbf{x}_{i,j}$, and then project them onto the eigenvector of $\mathbf{B}$ with the largest eigenvalue. Figure 13 shows the distribution of the resulting one-dimensional value, which now nicely separates the two colours that we want to distinguish, and which can be mapped onto greyscale values.

The preservation of phase information up to the PCA projection is essential for the success here. If, for comparison, we
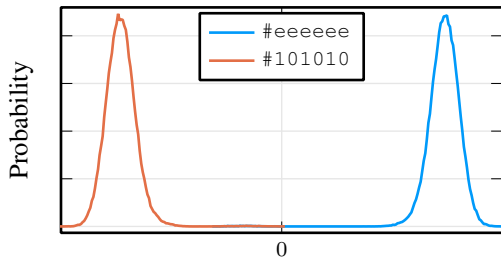
Fig. 13. Distributions of the 6-dimensional PCA projection of complex-valued pixels for two colours with complementary TMDS encodings from Fig. 8b
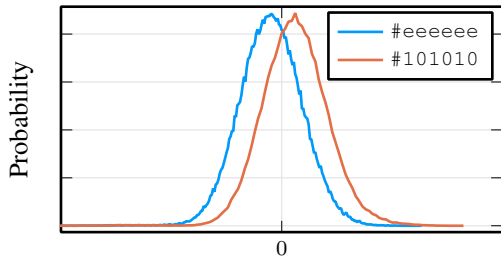

Fig. 14. Distributions of the 3-dimensional PCA projection of amplitude demodulated pixels for two colours with complementary TMDS encodings from Fig. 8c

instead first amplitude demodulate our samples, representing each pixel instead as a 3-dimensional vector

$$\mathbf{x}'_{i,j} = (|M_{i,3j}|, |M_{i,3j+1}|, |M_{i,3j+2}|) \in \mathbb{R}^3,$$

and perform PCA projection on those, then Figure 14 shows a much less clear separation of the two colours.

Figure 15 compares the colour-detection error rates achieved after either of these techniques, for both colour combinations discussed, again demonstrating that preserving the phase information significantly helps.

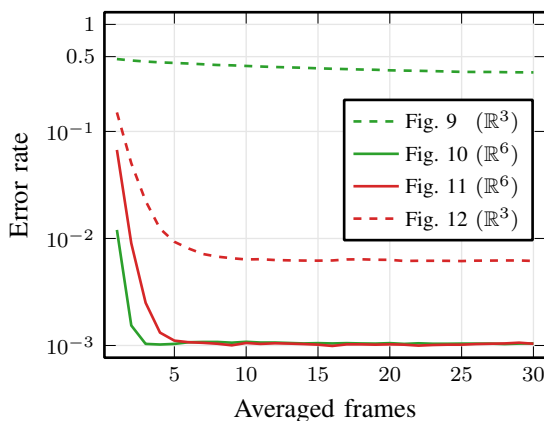Finally, Figure 16 uses 6-dimensional PCA on each line, showing again an improved contrast.

Fig. 15. Colour-detection error rates after 6-dimensional (solid) and 3-dimensional (dashed) PCA, for both colour combinations from Figs. 9–12 and differing number of averaged frames.
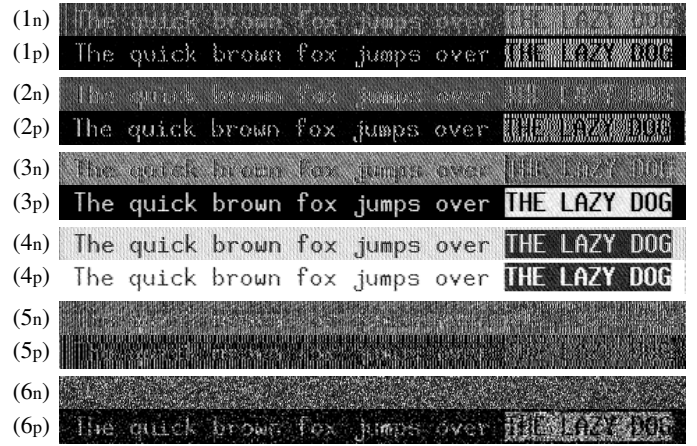

Fig. 16. Pairwise comparison of the first six lines from the test image. The top line in each pair is the same as in Fig. 7; the bottom line shows the PCA projection of 6-dimensional pixel data (three complex-valued samples) onto the eigenvector of $\mathbf{B}$ with the largest eigenvalue.

VII. CONCLUSIONS

The I/Q samples produced by SDR receivers carry both amplitude and phase information. We can more precisely estimate the target's pixel-clock frequency from its complex-valued autocorrelation sequence, using not only the position of a magnitude peak, but also its phase angle, and then frequency shift a harmonic of that pixel-clock frequency more precisely to 0 Hz. This way, we iteratively obtain a signal resampled at a very precise integer multiple of the pixel-clock frequency. This not only helps to correctly align the raster image, but also stabilizes the phase sampled for each pixel colour across many frames, enabling periodic averaging of complex-valued samples. The preserved phase information then also helps us to better discriminate between colours. In addition, if we have multiple samples per pixel, we can use dimensionality-reduction techniques to even further improve the contrast and pixel error rate.

REFERENCES

[1] W. van Eck, "Electromagnetic radiation from video display units: an eavesdropping risk?" *Computers & Security*, vol. 4, no. 4, pp. 269–286, 1985. https://doi.org/10.1016/0167-4048(85)90046-X

[2] M. G. Kuhn, "Electromagnetic eavesdropping risks of flat-panel displays," in *Privacy Enhancing Technologies*, LNCS 3424, Springer, 2005, pp. 88–107. https://doi.org/10.1007/11423409_7

[3] F. Elibol, U. Sarac, and I. Erer, "Realistic eavesdropping attacks on computer displays with low-cost and mobile receiver system," in *Proc. EUSIPCO*, 2012, pp. 1767–1771.
https://ieeexplore.ieee.org/abstract/document/6334179

[4] M. Marinov, "Remote video eavesdropping using a software-defined radio platform," Master's thesis, University of Cambridge, 2014.
https://github.com/martinmarinov/TempestSDR/

[5] C. O'Connell, "Exploiting quasiperiodic electromagnetic radiation using software-defined radio," Ph.D. thesis, University of Cambridge, 2019.
https://www.repository.cam.ac.uk/handle/1810/290902

[6] P. De Meulemeester, B. Scheers, and G. A. Vandenbosch, "Differential signaling compromises video information security through AM and FM leakage emissions," *IEEE Transactions on Electromagnetic Compatibility*, vol. 62, no. 6, pp. 2376–2385, Dec. 2020.