Focus: Robust Visual Codes for Everyone

Frederik Hermans*, Liam McNamara†, Gábor Sörös‡,
Christian Rohner*, Thiemo Voigt*†, Edith Ngai*
*Uppsala University †SICS Swedish ICT ‡ETH Zurich

ABSTRACT

Visual codes are used to embed digital data in physical objects, or they are shown in video sequences to transfer data over screen/camera links. Existing codes either carry limited data to make them robust against a range of channel conditions (e.g., low camera quality or long distances), or they support a high data capacity but only work over a narrow range of channel conditions. We present Focus, a new code design that does not require this explicit trade-off between code capacity and the reader's channel quality. Instead, Focus builds on concepts from OFDM to encode data at different levels of spatial detail. This enables each reader to decode as much data from a code as its channel quality allows. We build a prototype of Focus devices and evaluate it experimentally. Our results show that Focus gracefully adapts to the reader's channel, and that it provides a significant performance improvement over recently proposed designs, including Strata and PixNet.

1. INTRODUCTION

Codes that represent data through visual patterns provide a simple and proven means of wireless communication. For example, QR codes (Fig. 1a) are ubiquitously used to tag physical objects with digital data [8, 20] in applications including warehouse management, logistics, and supply chain management. Visual codes are also common in augmented reality scenarios, where they are used for relative positioning of the user [1] or to provide information to overlay on objects [18]. Recent work explored the use of streams of visual codes to transfer larger payloads from screens to hand-held devices over so-called screen/camera links [10, 13, 29, 39].

Although RFID is replacing visual codes in some applications, they remain attractive today. They are cheap and quick to produce, immune to radio interference, and do not end up as electronic waste. Furthermore, suitable readers are readily available almost everywhere: many people carry a camera-equipped smartphone with them.

MobiSys'16, June 25-30, 2016, Singapore, Singapore
© 2016 ACM. ISBN 978-1-4503-4269-8/16/06...$15.00
DOI: http://dx.doi.org/10.1145/2906388.2906399

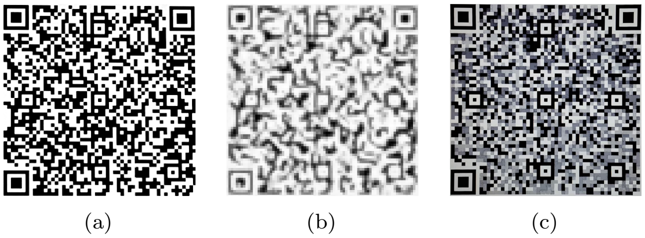


Figure 1: **Challenges of visual channels.** (a) Good channel: the QR code can be decoded. (b) Distance problem: the code is too blurred to decode. (c) Capture rate problem: a reader captures a stream of codes too slowly, resulting in an undecodable mix of two codes.

Challenge. Scanning visual codes, such as QR codes, can be a frustrating experience for consumers: the user may have to move very close to the code for scanning to succeed, only to find out that the code contains only a trivial amount of data. Short read distances, high error rates and low data capacity similarly hold back the usefulness of visual codes in industrial applications.

These issues are rooted in the high variance in quality of communication channels between readers and visual codes. In general, the quality of a visual channel is affected by the *distance* between the reader and the code (Fig. 1b) and by the reader's camera *resolution*. In the case of screen/camera links, the channel quality is also affected by the *rate* at which the reader captures frames [13]; too low capture rates cause readers to capture undecodable mixes of codes (Fig. 1c). These three factors—distance, resolution, and capture rate—vary widely between users and the devices they are using. Thus, a key challenge when designing visual codes is how to handle the wide range of channels over which users attempt to read them.

Existing codes handle this challenge in one of two ways. Either, they trade data capacity for increased robustness. For example, QR v1 codes can be read from long distances, but carry little data. Or, codes support high data rates, but make stringent requirements on the channel. For example, RDCode [39] requires the distance between code and reader to be within a few tens of centimeters, and PixNet [29] assumes capture rates of at least 60 FPS.

In this paper, we consider the design of high-capacity codes that can be fully decoded by readers with good channels, while at the same time readers with poor channels can extract essential data from them. Such codes would be uniquely useful. For example, a warehouse worker could get coarse-grained information about the items on a shelf even when reading the codes from a long distance, while he could obtain more detailed information when close-by. We discuss further application scenarios in §8.

Approach. We design Focus to address the stated challenge. A key idea of Focus codes is to partition a code's spectrum, i.e., its representation in the frequency domain, into many separate *sub-channels*. Each sub-channel encodes one part of the payload. By design, the sub-channels differ from each other in their robustness to the effects of distance and resolution. Furthermore, they can be decoded independently of each other. A reader who fails to decode one sub-channel may still decode other, more robust sub-channels.

Intuitively, each Focus sub-channel represents data with a different amount of *spatial detail*. Because distance and camera resolution affect a reader's ability to resolve detail (see Fig. 1b), the number of sub-channels that a reader can decode from a Focus code scales with its camera resolution and distance to the code. In particular, the sub-channels that use the least amount of detail can be decoded even by readers that are far away or that have poor cameras. Additionally, our design ensures that codes never contain more spatial detail than necessary to represent their payload, thereby improving the decodability over poor channels.

The use of independent sub-channels furthermore allows us to alleviate the *rate matching problem* on screen/camera links, which requires a transmitter's display rate to be matched to the readers' capture rates to avoid frame mixing (Fig. 1c). Instead of fixing the display rate, our *multi-rate streams* enable a transmitter to use different rates on different sub-channels. As a result, the transmitter can concurrently send data to readers with a variety of capture rates.

While Focus shares similarities with earlier work, namely Strata [14] and PixNet [29], it differs from them in important ways. Focus significantly improves on the performance of Strata by scaling more smoothly to the reader's channel and by ensuring that decoding errors cannot propagate between sub-channels. While PixNet targets high-quality DSLR cameras as readers, Focus is suited to a wide range of readers including smartphones, because it uses the spectrum more efficiently and has lower computational overhead.

Contributions. We make the following contributions:

- We present Focus, a new code design that is robust to common impairments of visual channels. Each reader can decode as much data from a code as its channel allows.
- We extend the design of our codes to support communication of large payloads over streams of codes on screen/camera links. We present *multi-rate* streams that enable a transmitter to concurrently support multiple readers with different frame capture rates.
- We experimentally evaluate a prototype implementation over a range of smart devices, display technologies, and channel conditions. The results show that Focus codes can be read at much longer distances and with significantly fewer errors than Strata [14] codes. On screen/camera links, Focus improves on the throughput of PixNet [29] by at least 3× for older smartphones and 2× for newer models; it improves on the communication range of RDCode [39] by 2× while providing superior goodput.

2. RELATED WORK

2D barcodes. There are many established 2D code designs, including QR [8], Data Matrix [16], and Aztec codes [17]. These codes are used in both professional [24] and consumer applications [20]. Recent designs like Microsoft's High Capacity Color Barcode [27] aim to improve the data capacity.

Of particular relevance for this work is Strata [14], a design that also represents data at different levels of spatial detail. Strata encodes payload in hierarchical, recursively-defined layers. The capacity and spatial detail in each layer grows exponentially, and thus requires an increasingly better channel for decoding. Focus differs from Strata in that it encodes data in the frequency domain; this enables Focus codes to contain less visual detail to represent the same amount of data. Furthermore, the level of spatial detail in a Focus code grows linearly between sub-channels, and their capacity is constant.

Visual OFDM. Encoding data in a code's spectrum rather than in blocks of pixels can be thought of as a visual variant of Orthogonal Frequency Division Multiplexing (OFDM). The concept of applying OFDM to visual codes was pioneered by Hranilovic et al. [12]. They provide a characterization of the link capacity of visual OFDM as well as an experimental validation of the idea.

PixNet [29] builds on visual OFDM to establish screen/camera links between LCDs and high-quality cameras, such as DSLR cameras with optical zoom, achieving throughputs of several MBit/s. A PixNet code consists of many small sub-codes, each of which is encoded with OFDM. A reader uses its knowledge about the spatial subdivision when correcting perspective distortion. The reader computes three Fourier transforms for each sub-code, which means that it computes hundreds of transforms for one captured code. In contrast, Focus codes are not spatially subdivided, but are exclusively partitioned in the frequency domain. They use less spatial detail and require a reader to only compute one Fourier transform for each captured code, thereby significantly reducing the computational overhead.

To the best of our knowledge, Focus is the first code design to bring the advantages of OFDM to smart devices with limited computational capacity and camera quality. Indeed, earlier work has claimed OFDM-based codes on smartphones to be infeasible [14], a claim that this paper refutes.

Barcode-based screen/camera links. Several 2D barcode designs have been proposed for screen/camera links [10, 22, 40]. RDCode [39] aims to improve the reliability of screen/camera links by designing custom error correction to handle device-specific limitations and impairments that arise from user behavior. In contrast, Focus builds on established Reed-Solomon and Fountain codes. LightSync [13] was designed to handle the rate mismatch problem, which leads to capturing a mix of two or more displayed codes. To that end, readers use per-line tracking to identify the constituent codes that make up a mixed code. In contrast, in Focus it is the transmitter that mitigates rate mismatch problems by using multi-rate streams (§4.3).

Recent efforts aim to embed data in video streams by using codes that cannot be noticed by human viewers [21, 34, 38]. They build on transmitters that support very high display rates (120 FPS), and currently require short distances between display and reader. The design of unnoticeable codes is out of the scope of this paper.

3. IMPAIRMENTS OF VISUAL CHANNELS

The quality of a visual channel depends on how accurately the reader's camera captures the displayed code. In this

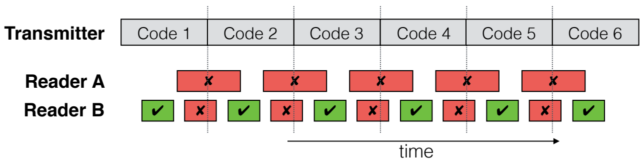


Figure 2: **Impact of frame capture rate.** Reader A captures mixes of two codes because its capture rate is too low. Reader B correctly captures each displayed code since its capture rate is twice the display rate.

section, we analyze three factors that have a strong impact on channel quality: the camera's *resolution*, the *distance* between camera and display, and the camera's *frame capture rate*. Our discussion of these impairments guides the design of Focus (§4, §5).

3.1 Impact of Camera Resolution & Distance

If a reader attempts to read a code from far away or with a very low resolution camera, the reader will *undersample* the displayed code. We now consider the impact of undersampling in the frequency domain.

Impact of too low resolution. Consider an idealized scenario where a camera is placed at a fixed distance from a display and is perfectly aligned with the display and focused on it. In this scenario, the camera's resolution effectively determines its spatial sampling rate Ω_c. Clearly, the camera's sampling rate has a crucial impact on whether a captured frame accurately represents the displayed code.

Let $\Omega_{\max}$ be the maximum frequency with a non-zero amplitude in the spectrum of the displayed code. Intuitively, $\Omega_{\max}$ corresponds to the size of the finest detail in the displayed code [36]. The Nyquist–Shannon sampling theorem states that if the camera's sampling rate Ω_c is $> 2\Omega_{\max}$, then the captured frame will accurately represent the displayed code. However, if $\Omega_c < 2\Omega_{\max}$, then the camera undersamples the displayed code and *foldback aliasing* occurs. In this case, the camera's sampling rate is too low to distinguish between the frequencies $\Omega_{\max}$ and $\Omega_f := \Omega_c - \Omega_{\max}$, the so-called foldback frequency [26]. Visually, the effect of undersampling is that crisp details in the displayed code are blurry in the captured code, as Fig. 1b illustrates.

A crucial observation that Focus builds on is that all frequencies below the foldback frequency remain unaffected by undersampling, i.e., the spectrum of the captured code matches the spectrum of the displayed code for frequencies $< \Omega_f$. This can also be seen in Fig. 1b, where the markers in the corners of the QR code—which are represented by relatively low frequencies—are better preserved than the small data-carrying blocks.

Impact of distance. The cameras in most smart devices do not have optical zoom. Therefore, increasing the distance to the displayed code implies shrinking the area of the code's projection onto the camera's image sensor. As a result, the effective spatial sampling rate decreases. However, as in the case of low camera resolution, the spectrum of the captured code will match the spectrum of the displayed code for all frequencies below the foldback frequency.

3.2 Impact of Frame Capture Rate

Frame mixing is a common issue when streams of codes are used to transfer data over screen/camera links [13, 22, 29]. The problem arises if the transmitter's display rate and a reader's capture rate are misaligned as illustrated in Fig. 2. In the shown scenario, Reader A captures frames at the same rate as the transmitter displays codes. Because the exposure time of each captured frame overlaps with the time at which the transmitter updates the displayed code, every frame that Reader A captures contains two codes overlaid on each other. An example of two mixed codes is shown in Fig. 1c. In general, mixed codes cannot be decoded.

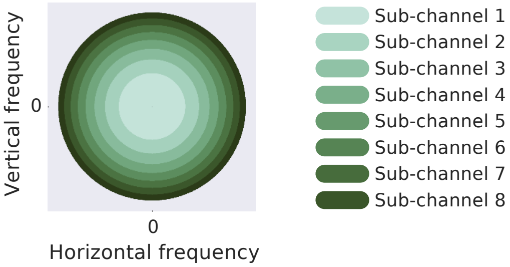


Figure 3: **Partitioning of a Focus code's spectrum into sub-channels.** Sub-channels with lower indices contain lower-frequency elements of the spectrum, which are more robust to undersampling.

The problem does not affect readers whose capture rate is at least twice as large as the display rate, such as Reader B in Fig. 2. In this case, at least every other captured frame will contain only one displayed code. However, setting the display rate to be half the slowest reader's capture rate is an unsatisfying solution, because frame capture rates vary widely between readers. For example, an Apple iPhone 6 can capture up to 240 frames per second (FPS), whereas we observe a capture rate of around 5 FPS for a 2010 Google Nexus One. Thus, a transmitter that displayed codes at a rate of 2.5 FPS would leave the channel of an Apple iPhone 6 vastly underutilized.

3.3 Other Impairments

In practical settings, there are additional factors that may affect the quality of a visual channel, such as unfavorable lighting, perspective distortion, lens distortion, or blur. Focus readers correct for perspective and lens distortion, but do not employ explicit measures against other channel impairments. However, we experimentally study the impact of lighting and perspective distortion in the evaluation.

4. FOCUS CODES

With the given background on channel impairments, we now present the core ideas behind Focus codes. We concentrate on the ideas in this section, and consider the complete design of a transmitter and a reader in §5.

4.1 Code Construction

Rather than encoding data directly into spatial blocks of pixels, as most visual codes do, Focus encodes data in a code's spectrum. This means that for a given payload we construct the complex spectrum S of the code, and then apply the inverse Fourier transform to produce a grayscale image, which is the actual code.

First, we conceptually partition the spectrum into *sub-channels*. Note that an element $S[u, v]$ of the spectrum describes a sinusoid with frequency $\sqrt{u^2 + v^2} \cdot 2\pi$. Higher frequencies correspond to finer spatial detail. We define a set of sub-channels on the spectrum as follows. Let $s_1, s_2, s_3, \ldots$ be the sequence of elements of S ordered by increasing frequency. We define the first sub-channel to contain the first k elements of this sequence, the second sub-channel to contain the next k elements, and so on. The choice of k defines the data capacity of each sub-channel. Fig. 3 shows the spectrum partitioning of an example code with 8 sub-channels. By construction, the first sub-channel contains the elements with the lowest frequencies. Thus, it corresponds to gradual changes in light intensity in the spatial representation of the code, and it uses the least amount of spatial detail. Each subsequent sub-channel uses more spatial detail. The last sub-channel defines the code's finest details.

Next, we encode the payload in the sub-channels. The payload bytes are modulated using quadrature phase-shift keying (QPSK), yielding a sequence of complex symbols. Each symbol encodes two bits of payload data in its phase. We load the first k symbols into the first sub-channel, the next k symbols into the next sub-channel, and so on. Because each sub-channel describes a subset of elements of the spectrum, we are essentially populating the spectrum with our payload symbols. During this process, we ensure that the resulting spectrum observes Hermitian symmetry, so that its inverse Fourier transform is real-valued.

Finally, we compute the inverse Fourier transform of the spectrum. The result is a real-valued matrix, which we can interpret as a grayscale image. This grayscale image is the Focus code that corresponds to the given payload. A reader that captures the code can recover the payload by computing the code's Fourier transform and demodulating the symbols from the sub-channels.

4.2 Robustness to Undersampling

There are two reasons why Focus codes are particularly robust to undersampling caused by long distances or low camera resolution.

First, a Focus code uses only the lowest frequencies that are necessary to represent its payload. For example, if the entire payload fits into the first sub-channel, then the other sub-channels will not carry any data. Consequently, a Focus code never contains more spatial detail than necessary. This is important, as an image's finest details are most prone to undersampling. Therefore, Focus codes can be read over longer distances or with worse cameras than codes that encode data in higher frequencies, such as the high-contrast edges that are characteristic of most barcodes.

Second, if a Focus code is undersampled and aliasing occurs, the *later* sub-channels will be affected, since their symbols are represented by the highest frequencies. Crucially, due to the nature of aliasing, the sub-channels whose elements are represented by frequencies less than Ω_f are unaffected. A reader can still correctly decode the data from the unaffected sub-channels[1]. Furthermore, decoding errors in one sub-channel do not cause decoding errors in other sub-channels, because sub-channels are independent from each other. The independence follows from the fact that the symbols in the sub-channels correspond to elements in the spectrum, which in turn are coefficients of basis vectors. These are independent of each other by definition. The partitioning of the code's spectrum into sub-channels thus enables partial decoding, which in turn ensures that the amount of data a reader can decode from a Focus code scales with the amount of undersampling—the less a code is undersampled, the more data can be decoded.

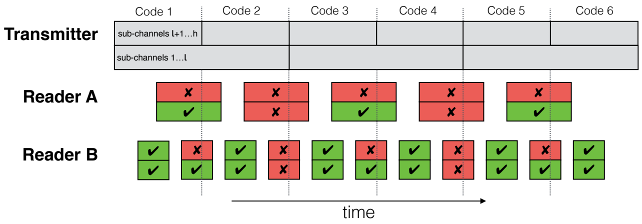


Figure 4: **Multi-rate streams alleviate the problem of mixed frames.** Reader A only captures mixed frames because its capture rate is too low, but it can nonetheless successfully decode data from the stream.

4.3 Robustness to Frame Mixing

Focus's independent sub-channels allow us to alleviate the frame mixing problem on screen/camera links. We design *multi-rate streams* that use different data rates across the sub-channels to enable a transmitter to simultaneously accommodate readers with different capture rates.

The concept is best explained by an example. If a transmitter wants to support readers with captures rate of 15 FPS and 30 FPS, it constructs a multi-rate stream with a total of $h + l$ sub-channels per code. The transmitter then updates the l sub-channels at 7.5 FPS, i.e., for every other code it displays. And it updates the h sub-channels at 15 FPS, i.e., for every code. Fig. 4 illustrates the situation.

The capture rate of a 30 FPS reader (Reader B) is twice the display rate. Thus, Reader B is guaranteed to correctly capture every code the transmitter displays, and can decode the data from all sub-channels. In contrast, every frame captured by a 15 FPS reader (Reader A) contains mixed codes. However, because the data in the first l sub-channels changes only in every other code, Reader A can nonetheless decode data from every second mixed frame it captures.

To see how this works, we can model a mixed frame c_{mix} as a linear combination $c_{\text{mix}} = ac_i + bc_{i+1}$ of two codes c_i, c_{i+1}, where a and b are attenuation factors. Assume that the data on the first l sub-channels did not change between c_i and c_{i+1}. This means that the spectra S_i, S_{i+1} of c_i and c_{i+1} are identical in the first l sub-channels. Now, for any symbol $S_{\text{mix}}[u, v]$ in the first l sub-channels of the mixed code:

$$\begin{aligned}
S_{\text{mix}}[u, v] &= \mathcal{F}(c_{\text{mix}})[u, v] = \mathcal{F}(ac_i + bc_{i+1})[u, v] \\
&= aS_i[u, v] + bS_{i+1}[u, v] \\
&= (a + b)S_i[u, v] = (a + b)S_{i+1}[u, v].
\end{aligned}$$

The second line holds due to the linearity of the Fourier transform. The third line follows because the symbols in the first l sub-channels are identical. The above equation implies that frame mixing only scales the magnitude of identical symbols by $(a + b)$. Crucially, the phase is unaffected. Since Focus encodes data only in the phase, the 15 FPS reader can decode the data from the first l sub-channels, as illustrated in Fig. 4. This simple example can be readily extended to readers with a range of different capture rates. However, transmitting the same payload at different rates on different sub-channels would lead to an underutilization of

[1] Note that the reader does not need to know the value of Ω_f. It is enough for the reader to verify if a sub-channel was decoded correctly, e.g. by using a checksum.

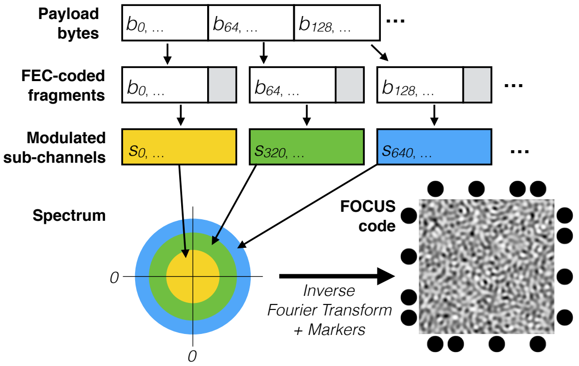


Figure 5: **Construction of a Focus code.** By design, a Focus code represents different parts of the payload by different amounts of spatial detail.

high-rate readers, since they would receive many data blocks multiple times in both the higher and the lower sub-channels. Therefore, Focus uses Fountain coding [35] across the sub-channels to make sure that high-rate readers can use the data from all sub-channels. We describe the use of Fountain codes in §5.3.

5. DESIGN

Based on the description of our core ideas, we now describe the complete encoding and decoding process, as well as the construction of streams of Focus codes.

5.1 Encoding

Fig. 5 provides an overview of how a Focus code is constructed. For simplicity, we refer to the entity that generates a code as the *transmitter*, even though the transmitter may actually simply print a code on paper.

Fragmentation and forward error correction. Transmitters split the payload bytes into *fragments* of 64 bytes each. Each of the payload fragments is later encoded in one sub-channel. Due to effects such as sensor noise, even a reader that samples a code at a sufficiently high rate may sporadically decode some bytes incorrectly. Therefore, the transmitter applies a Reed-Solomon error correcting code [32] to each fragment. Since we aim to support a wide range of channels, the transmitter conservatively adds 16 parity bytes to each fragment. Thus, a coded fragment is 80 bytes long, and readers can recover up to 8 byte errors per fragment. Note that error correction is applied to individual fragments rather than to the payload as a whole. This enables a reader to decode data from all fragments that have eight errors or fewer. Note also that redundancy in the code is effectively spread throughout the spatial representation of the code.

Modulation and spectrum construction. Because Focus represents data in the frequency domain, the coded fragments need to be modulated into complex symbols. To ensure the decodability of multi-rate streams, only the phase of a symbol is used to carry data. We use QPSK as the modulation scheme, so each symbol represents two payload bits. We empirically found QPSK to give a good trade-off between code capacity and robustness. Each modulated fragment is encoded in one sub-channel and consists of $k = 320$ symbols (= 80 bytes × 4 symbols/byte).

Next, the transmitter constructs the code's spectrum. The spectrum is a complex-valued matrix whose entries are initially all zero. The symbols of the first sub-channel are copied to the entries around the origin of the spectrum, as shown in Fig. 5. The symbols of the second sub-channel are arranged in the second ring around the center, and so on. During the construction process, the transmitter ensures that the resulting spectrum observes Hermitian symmetry through the origin [12]. This is necessary to ensure that the spectrum's inverse Fourier transform is real-valued, and thus can be displayed as a grayscale image. To ensure symmetry, the transmitter only populates the upper half of the spectrum, and sets the lower half to be a conjugate point reflection around the origin.

Inverse Fourier transform and clipping. Next, the transmitter computes the inverse Fourier transform of the constructed spectrum. The result is a real matrix that can be displayed as a grayscale image. However, the image may have a high peak-to-average ratio (PAR), a well-known issue in OFDM [31]. Due to finite levels of intensity in the output image, a high PAR in the real matrix causes quantization errors. To reduce the PAR, transmitters employ a standard technique of clipping the real matrix's maximum value [31].

Location markers. Finally, the transmitter adds markers around the code which help the reader to locate the code in a captured frame. We use a simple set of 16 filled circles as markers, as they enable a reader to locate a code with very high accuracy, as we describe in the next section. The robust localization of Focus codes without external markers is part of our ongoing research.

5.2 Decoding

We now describe how a reader decodes data from a code in a frame that it captured.

Correcting lens distortion and locating a code. As a first step, the reader corrects the captured frame for lens distortion. Radial lens distortion maps straight lines in a scene to bent lines in the captured frame, and is very common for the cheap lenses used in smart devices. We use standard techniques for undistortion [19]. Our experience shows that this correction step is crucial for good decoding performance at short distances.

Next, to locate a code in a captured frame, the reader looks for the 16 circular markers, which may appear as ellipses due to perspective distortion. The reader then computes the geometric center of each marker. We use circular markers because their centers are invariant to perspective distortion and can be computed with sub-pixel accuracy [4].

Correcting perspective distortion. Perspective distortion occurs if the reader views the code at an angle. As a result, a square code may appear as a more general quadrilateral in a captured frame. To compute the Fourier transform of a code, the reader must first undo the effect of perspective distortion. To this end, the reader determines the affine transform that describes the distortion. The transformation matrix can be constructed from the location of the markers in the captured frame, and knowledge about the location of markers in an undistorted frame. Then, the reader resamples the distorted code into its original form.

Locating and resampling a code in a captured frame can be viewed as achieving spatial synchronization with the transmitter. Just as in traditional OFDM systems, synchronization errors cause severe inter-symbol interference and poor decoding performance [2, 30, 33]. Therefore, the reader must

Model	Resolution		Capture rate
	Photo	Video	
Samsung Galaxy S6 (2015)	16 MP	2 MP	30 FPS
Samsung Galaxy S3 (2012)	8 MP	2 MP	30 FPS
Google Nexus One (2010)	5 MP	0.3 MP	5 FPS
Google Glass (2014)	5 MP	0.9 MP	30 FPS

Table 1: **Reader devices used in the evaluation.**

determine the projection matrix with high accuracy. This is why the sub-pixel localization accuracy of our markers is especially important.

Fourier transform, demodulation, and error correction. Finally, the reader computes the Fourier transform of the undistorted code. It unloads the code's spectrum, reversing the process described in §5.1. The reader then demodulates the symbols into bytes, and corrects errors using the Reed-Solomon code. All recovered and error-free fragments are passed to the application.

5.3 Streams of FOCUS Codes

Streams of Focus codes are used to communicate data of arbitrary size on screen/camera links. We briefly describe the construction of multi-rate streams and the use of Fountain codes in Focus.

Multi-rate streams. The construction of Focus's multi-rate streams is a simple generalization of the example from §4.3. Assume the transmitter would like to support readers with capture rates of $r_1, r_2, \ldots, r_n$, where n is at most as large as the number of sub-channels in the codes. Before the transmission starts, the transmitter chooses for each r_i a subset of sub-channels on which it transmits at a rate of $\frac{1}{2}r_i$. More specifically, if the transmitter's display rate is d, the data on the chosen sub-channels changes only on every $2\frac{d}{r_i}$-th code that the transmitter displays.

It is up to the system designer to decide which capture rates should be supported and how many sub-channels are assigned to each capture rate. The larger the number of sub-channels assigned to a capture rate, the higher the throughput for readers with that rate. In practice, we suggest that most sub-channels are assigned rates that are suitable for the most common capture rates (e.g., 30 FPS), a few are assigned higher rates (e.g., 60 FPS) to boost throughput for high-end smart devices, and few transmit at low rates (e.g., 2 FPS) to ensure that even legacy devices can receive data.

Fountain erasure coding. Screen/camera links are commonly uni-directional, and so a transmitter does not know which parts of the payload the readers have already successfully decoded. This makes it difficult to decide which parts of the payload should be re-sent, and on which sub-channels.

We avoid the issue by using Raptor erasure codes [35], a state-of-the-art Fountain code [6]. To transmit a payload consisting of n data blocks, the transmitter uses the Raptor code to produce a (infinite) sequence of fragments, which are then encoded in the sub-channels of the stream's codes. A crucial property of Raptor codes is that a reader only needs to receive *any* $n + \epsilon$ of the transmitted fragments to recover the full payload. Thus, it does not matter which fragment is encoded in which sub-channel, and there is no need for explicit synchronization between reader and transmitter. Furthermore, Raptor coding enables us to avoid sending the same payload in sub-channels that use different rates. Instead, every fragment that a reader decodes provides useful progress towards obtaining the full payload.

6. EVALUATION OF STATIC CODES

We now present our evaluation of a prototype implementation of Focus[2]. The prototype is written in Python and Java and runs on both Android devices and regular computers. We make heavy use of OpenCV [5] for image processing, and use FFTW [9] for computing Fourier transforms. We have further optimized critical code sections using ARM NEON instructions, and our implementation supports real-time decoding even on handheld Android devices.

We use four different reader devices in the evaluation that differ in their camera quality: a Samsung Galaxy S6, which is a state-of-the-art smartphone; a Samsung Galaxy S3, which is an older, yet popular model; a Google Nexus One, which represents a legacy smartphone released in 2010; and Google Glass as a representative for wearable technology. Key information about the devices is summarized in Tab. 1.

In this part of the evaluation, we study the decoding performance of static Focus codes. To capture static codes, readers take high-resolution photos of the codes, rather than videos. The readers are placed on a tripod except for our experiments on the impact of hand motion. The displayed codes are 20×20 cm^2 large. We vary the distance between readers and codes to assess the impact of code size.

6.1 Impact of Camera Quality

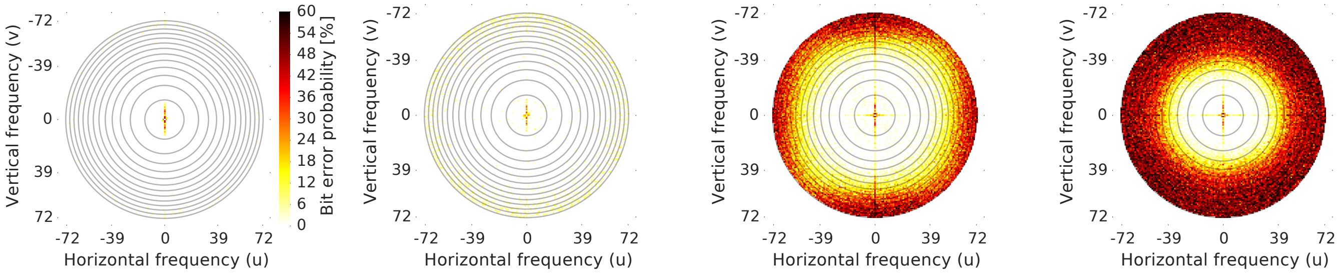
We begin by testing the claim that a reader's decoding performance scales with its camera quality. To get a detailed insight, we consider the bit error rates of different sub-channels as a measure of decoding performance. We expect an increase in error rates for later sub-channels, particularly on low-end cameras.

Experimental setup. We place the four readers at a distance of 3 m from an LCD at no angle. The LCD shows a Focus code carrying a payload of 1.6 KB. For each reader, we repeat the experiment 20 times with different code payloads.

Results. Fig. 6 shows the average bit error rates (BER) in the spectra of codes captured with the Galaxy S6, Galaxy S3, Nexus One, and Google Glass. A low bit error rate is denoted by white, a high error rate is denoted by red. The gray circles indicate the sub-channels (cf. Fig. 3). We observe only little variance in error rates between codes, and thus do not show variance in the plot.

The figure shows that bits encoded in lower frequencies generally have lower error rates than bits encoded in higher frequencies. The first sub-channel, closest to the origin, poses an exception that we address in the next paragraph. We can also see that there are more error-free channels for readers with better cameras. The plot for the Galaxy S6 is virtually error free (Fig. 6a) except for the first sub-channel. For the Galaxy S3 (Fig. 6b), we observe error rates of ca. 5% for very high frequencies. For the Google Nexus One (Fig. 6c), we see an increase in errors in later sub-channels, but there are still several sub-channels with error rates < 5%. Google Glass's camera does not support autofocus and is by far the worst camera in our experiment, but nonetheless a few sub-channels that use very low frequencies have only few errors (Fig. 6d).

[2] Focus is available as open source under `https://github.com/frederikhermans/focus`.



(a) Galaxy S6 (b) Galaxy S3 (c) Nexus One (d) Google Glass

Figure 6: **Location of bit errors in the frequency spectrum of static Focus codes for different readers.** Lower frequencies (closer to the center) are more robust to bit errors.

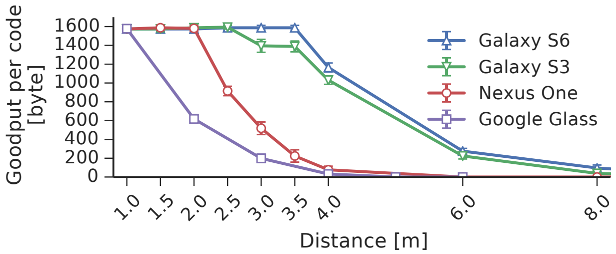


Figure 7: **Successfully decoded payload bytes as a function of distance to the code.** Each reader's goodput scales smoothly with its distance to the code.

Some payload symbols in the first sub-channel, i.e., the one closest to the spectrum's origin, have high BER, as evidenced by the small cross in the origin. Our investigation showed that this is an effect of the LCD's refresh, which manifests as a dark bar across the captured frame. The effect can be observed for all LCDs, but disappears when codes are printed on a sheet of paper. Since few bits are affected, the errors can be recovered by error correction.

We conclude that Focus's basic premise is correct: lower frequencies provide a higher robustness to bit errors. By encoding data in sub-channels, even a reader with a poor camera, such as Google Glass, can at least partially decode a Focus code. The better the reader's channel in terms of camera quality, the more data it can decode from the very same code.

6.2 Impact of Distance

We now consider how decoding performance changes with the distance between reader and display. Distance has a major impact on channel quality, because the area of the captured code decreases as the distance between reader and display increases. The following experiment also allows us to gauge the impact of the (spatial) code size. The effect of increasing the distance to a code of fixed size is similar to the effect of reducing the code size while keeping the distance constant—both reduce the size of the code's projection on the reader's image sensor.

Experimental setup. An LCD screen shows a Focus code with a payload of 1.6 KB. We place the readers at distances ranging from 1 m to 10 m from the display and measure the goodput and the bit error rate. We define the goodput as the number of correctly decoded bytes (after error correction) excluding parity bytes. For each reader, we capture 20 codes with different payloads and present averages and their respective standard deviations.

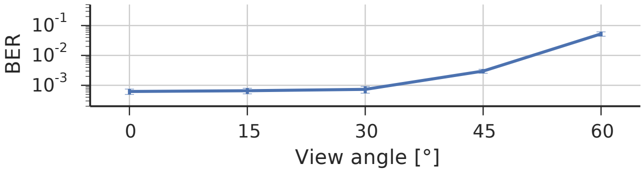


Figure 8: **Impact of view angle between camera and display on the bit error rate.** Perspective distortion has only little impact on decoding performance.

Results. Fig. 7 shows the goodput for the different readers as a function of distance to the display. The distance at which a code can be completely decoded depends on the reader's camera quality. While the Galaxy S6 can decode all data at a distance of 3.5 m, Google Glass needs to be at a distance of 1 m from the display. For all readers, the goodput decreases smoothly (rather than abruptly) as the distance increases. This graceful degradation is an effect of the independence of sub-channels. These observations demonstrate how the decoding performance adapts to the reader's channel quality. We also note that the variance in goodput is very low. This suggests that the decoding performance is independent of the code's payload; i.e., there are no codes that are particularly difficult to decode. We conclude that the goodput scales smoothly with the reader's channel quality in terms of distance to the code.

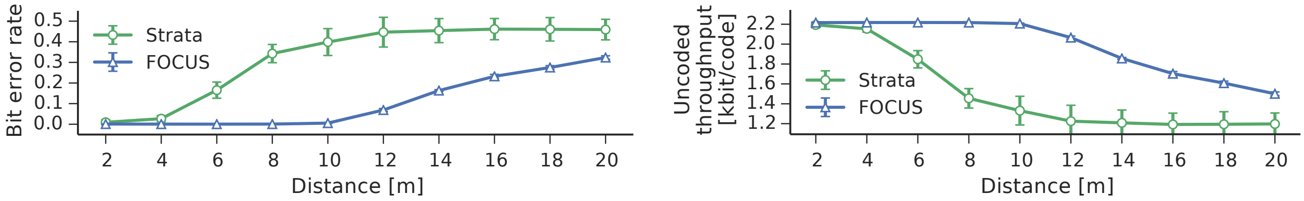
6.3 View Angle, Lighting, Display Medium, and Hand Motion

Next, we briefly consider the impact of view angle, ambient lighting, display medium, and hand motion in terms of BER.

View angle. If the reader views a code at an angle, the captured code will be subject to perspective distortion. To understand the impact of perspective distortion on the decoding performance, we perform an experiment in which the Galaxy S6 is placed at a distance of 2 m from an LCD displaying Focus codes with a capacity of 1.6 KB. Between experiments, we vary the angle at which the reader views the display from 0° to 60°.

Fig. 8 shows the average BER we observe for different view angles. Even at a view angle of 60°, the BER is only 5%, so errors can be corrected by Reed-Solomon coding. For angles less than 45°, the impact of perspective distortion is barely noticeable. We conclude that Focus codes are robust to perspective distortion.

Lighting. We next consider the impact of ambient lighting. Variable lighting conditions cause reader cameras to perform various adaptations, such as with exposure time. We perform



(a) Bit error rate

(b) Uncoded throughput

Figure 9: **Performance of Strata and Focus as the reading distance increases.** Focus codes have lower bit error rates (left) and deliver more data (right) than Strata codes across all distances in our experiment.

an experiment similar to the previous setup, but we fix the view angle at 0°, distance at 2 m, and vary the lighting conditions from darkness to full neon strip lights. We also perform the experiment outdoors. Our results showed BERs below 0.06%, so we conclude that ambient lighting does not have an appreciable impact.

Ambient lighting does not impair the decoding performance of Focus codes because it only affects the average light intensity of a captured code. As such, it changes only the DC component of the code's Fourier transform [29], which does not encode any data. Note that a shadow cast over a Focus code will also affect only the average light intensity of the captured code, and therefore is not expected to significantly impact bit error rates either.

Display medium. Focus codes can be displayed on a variety of media. We now consider the impact of the display medium on decoding performance. We display a low-capacity Focus code (384 bytes) on an LCD monitor, an LED monitor, a 3rd generation Apple iPad, and a sheet of paper. The printed code was printed with an office laser printer. We use the Galaxy S6 as the reader and place it at distances from 30 cm to 150 cm from the codes.

The receiver decoded the complete payload from all media and at all distances. Our inspection of the BER did not reveal any medium-specific characteristics. This is particularly encouraging when considering that laser printers produce grayscales by varying the number of black dots in an area (halftoning), since they can only produce monochrome output. Our result suggests that even codes reproduced with halftoning can be successfully decoded. We conclude that Focus codes are suitable for a range of display media.

Hand motion. In the experiments we have presented so far, the readers were placed on tripods to ensure repeatability and to simplify experimentation. In a real-world scenario a user may hold a reader in their hands instead. In this case, the reader will be subject to inevitable minor tremors and movements. These may cause motion blur in the captured image if there is movement during the exposure, and it may make it hard to align the reader's image plane center with the code's center, in particular at longer distances.

Our experience with the Focus prototype during a live demo [11] suggests that minor movements of the reader do not degrade decoding performance. To test this more systematically, we repeat a subset of the experiments described in §6.2. A user holds the Samsung Galaxy S6 in their hand and decodes ten codes at distances of 2 m, 4 m, 5 m, and 6 m from the display. We then compare the BER of codes captured with a handheld reader with the BER when the reader is placed on a tripod. We found that the difference in BER is very low (< 3%) across all distances. In particular, we did not find the BER for handheld readers to increase with the distance to the code. We conclude that minor movements, which are inevitable if a reader is handheld rather than statically positioned, do not have a significant impact on Focus's performance.

6.4 Comparing FOCUS and Strata

We now compare Focus to Strata [14], a recently proposed 2D code that shares Focus's goal of supporting different channel qualities. Strata codes are comprised of several spatial layers whose purpose is similar to that of sub-channels in Focus. In this experiment, we investigate which code can deliver more data on a given channel, and which code adapts more gracefully to a decrease in channel quality. Because Strata does not specify its error correction parameters, and in order to get a general view of the performance, we focus on the bit error rate (BER) as the primary metric for the experiment. We have been unable to obtain a copy of Strata. Therefore, we have implemented Strata ourselves based on the paper [14].

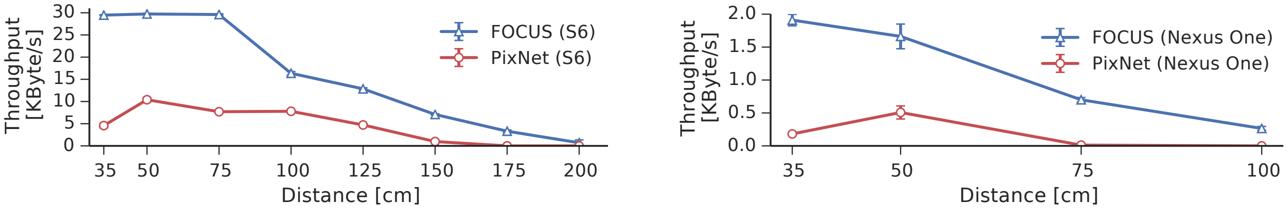
Experimental setup. We construct four-layer Strata codes with a capacity of 2267 bits and Focus codes with identical capacity and dimensions. The codes are displayed on an LCD and we use the Galaxy S6 as the reader. We vary the distance between reader and display from 1 m to 20 m. At each distance, we display 10 codes of each type and compute the average BER and the average uncoded throughput as well as the respective standard deviations.

Results. Focus codes have a lower bit error rate than Strata codes over all distances (Fig. 9a). Furthermore, Focus's bit error rate is close to zero for distances less than 12 m, and then grows slowly[3]; in comparison, Strata's bit error rate already exceeds 15% at a distance of 6 m, and then grows much more quickly. Focus provides a 3× longer read range than Strata (12 m vs. 4 m), if we assume the tolerable error rate to be 10%.

Fig. 9b shows the uncoded throughput per code for Focus and Strata, which we define as the number of bits in the decoded payload that match the transmitted payload, without any error correction. Note that the uncoded throughput has a lower bound of $\frac{1}{2}\times$ code capacity, as a reader that simply guessed each bit would be expected to guess half of the transmitted payload bits correctly. The plot shows that Focus delivers more bits correctly than Strata across all distances in our experiment.

One reason for Strata's larger BER (and hence its lower throughput) is that it uses more spatial detail to represent data than a Focus code of comparable capacity. Strata organizes payload into recursively-defined layers, and most of the payload is encoded in the deepest layer. This deepest

[3] We reach longer distances in this experiment compared to §6.2 because the code's payload is considerably smaller.



(a) Samsung Galaxy S6 reader

(b) Google Nexus One reader

Figure 10: **Throughput achieved by Focus and PixNet over distance for (a) a recent and (b) an old smartphone.** Focus improves the throughput by a factor of at least 2× for all distances and both devices.

Task	Duration (ms)	Relative
Correct lens distortion	7 ms	13%
Locate code	10 ms	19%
Perspective transform	4 ms	7%
Fourier transform	20 ms	37%
Demodulation	2 ms	4%
Error correction	2 ms	4%
Total	54 ms	100%

Table 2: **Decoding overhead breakdown for one frame on an S6.** By parallelizing the decoding of multiple frames, our prototype can decode at a rate of 56 FPS.

layer represents data in fine spatial details that are $\frac{1}{64}$th of the code's width. This explains the steep increase in BER as the distance to the display increases. In contrast, the finest detail that a Focus code uses to display a 2267 bit code is about $\frac{1}{26}$th of the code's width. This makes them more robust to undersampling.

Furthermore, decoding errors can propagate to deeper layers in Strata. This happens if the orientation of a block is incorrectly detected. We attribute the large performance variance of Strata to the propagation of decoding errors. In contrast, sub-channels in Focus codes can be decoded independently of each other, and we observe very little variance. We conclude that Focus codes enable a reader to make better use of their channel capacity, as they deliver data with lower bit error rates.

7. EVALUATION OF STREAMS OF CODES

We now evaluate streams of Focus codes that are used to transmit large payloads on screen/camera links. This scenario differs from the decoding of static codes because the displayed codes change rapidly. Furthermore, the reader's resolution of video frames is usually much lower than the resolution of photos (cf. Tab. 1).

7.1 Microbenchmarks

A reader on a screen/camera link must process its captured frames at a high speed to achieve a satisfying throughput. We briefly characterize the processing speed of our prototype reader on the Galaxy S6. We measure the time that it takes on the S6 to decode 900 captured frames, i.e., 30 seconds of video captured at 30 FPS. The multi-threaded prototype takes 15.9 s for this task, which corresponds to an effective processing rate of 56.7 FPS. This means that our prototype supports real-time decoding of frames that are captured at a rate of 56 FPS or less. A breakdown of a single thread's decoding overhead of one frame is provided in Tab. 2.

7.2 FOCUS and PixNet

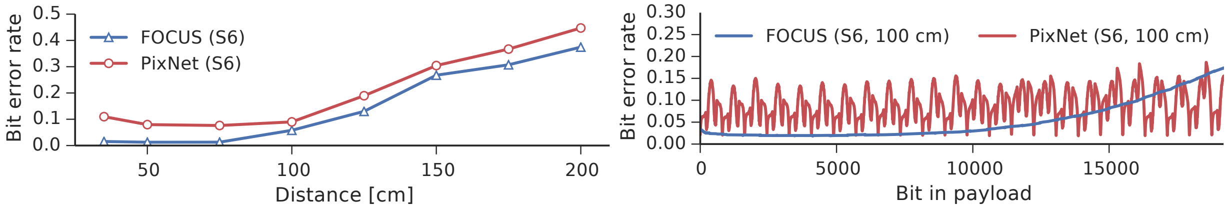
We now consider Focus's throughput under different channel conditions. For reference, we also measure the throughput of PixNet [29], an earlier OFDM-based code for screen/camera links. We have been unable to obtain a copy of PixNet, and thus have implemented it based on a study of the relevant papers [28, 29].

We define the throughput as the number of correctly received bytes (*after* error correction) per second. Duplicate chunks of data (i.e., identical chunks that have already been received earlier) do not count towards the throughput: if a receiver captures the same code from a transmitter's stream twice, any data it decodes again from the second capture does not increase the throughput. While our definition of throughput is rather strict, we believe that it provides a better understanding of the rate at which a reader can obtain useful data. We also briefly consider the bit error rates of Focus and PixNet to obtain a better low-level characterization of the respective channels.

Experimental setup. An LCD shows a stream of either Focus or PixNet codes. The individual codes in both streams have the same size and capacity (2 KB). We use the Galaxy S6 and the Nexus One as readers to represent high- and low-end readers, respectively. The transmitter's display rate is set to half the reader's capture rate. We vary the distance between display and reader from 35 cm to 200 cm. Each experiment run is repeated six times, and we present averages and their respective standard deviations.

Throughput results. Fig. 10a shows the throughput for the Galaxy S6. PixNet's throughput peaks at 10 KB/s at a distance of 50 cm. It then gradually falls to zero at 175 cm. We attribute the poor throughput at 35 cm to the fact that PixNet does not correct lens distortion. Focus achieves the maximum achievable throughput (30 KB/s) for distances up to (including) 75 cm. At larger distances, its throughput gradually declines until it reaches 0.6 KB/s at a distance of 200 cm. Over all distances, Focus's throughput is at least twice as high as PixNet's. Fig. 10b shows the throughput for the Nexus One. In this experiment, the maximum achievable throughput is 4 KB/s due to the transmitter's reduced display rate. While both Focus and PixNet fall short of the maximum throughput, Focus provides an improvement of at least 3.3× over all distances.

There are several reasons for Focus's superior throughput. First, Focus uses lower frequencies to represent data than PixNet. PixNet divides its codes spatially into many smaller sub-codes, each of which then encodes part of the payload using OFDM. Note that the lowest frequency in an OFDM code is determined by the size of the code. As a result, most data in a PixNet code is represented by similar, relatively high frequencies. In contrast, Focus codes are only divided



(a) Bit error rate over distance

(b) Bit error rate within the payload at a distance of 100 cm

Figure 11: **Bit error rates for Focus and PixNet over distance and within the payload.** (a) Focus's average bit error rate per frame is consistently lower than PixNet's; (b) through its careful arrangement of the payload in the frequency domain, Focus ensures that errors are concentrated toward the end of the payload, enabling effective error correction.

in the frequency domain and so use lower frequencies with a wider frequency range to represent data. Second, unlike PixNet, Focus readers correct for lens distortions that are common on smart devices. Third, PixNet discards all mixed frames, whereas Focus readers are able to decode mixed frames. Finally, since PixNet targets good channels, it uses relatively weak error protection for some parts of the payloads. We believe that while PixNet is well-suited for the high-end cameras it targets, Focus provides superior performance on smart devices.

BER results. Throughput is an important metric for developers who want to understand which system delivers more useful bytes per second. To obtain a more fine-grained characterization of the communication channels provided by Focus and PixNet, we now consider the respective bit error rates.

Fig. 11a shows the average bit error rate over distance for the Samsung Galaxy S6. Across all distances, Focus achieves a lower bit error rate than PixNet. However, it may be surprising that the difference in bit error rate is not as large as the difference in throughput. This is because Focus readers try to decode mixed frames, rather than discard them as PixNet does. The advantage of decoding mixed frames is that it relaxes the need for tight temporal synchronization between the transmitter and the reader (§4.3). If the mixing is not too severe, or if the transmitter uses multi-rate streams, a Focus reader can partially decode mixed frames. However, a mixed frame will generally have a higher bit error rate than a clean frame. Thus, because Focus considers mixed frames whereas PixNet discards them, Focus's average bit error rate is relatively high in Fig. 11a.

It is instructive to consider how errors are distributed within the payload. Fig. 11b shows the bit error rate as a function of bit position in the payload for a distance of 100 cm between reader and transmitter. PixNet's bit error rate is similar across the whole payload. The pattern in its curve is an artifact of how PixNet arranges data in the frequency domain. In particular, the valleys correspond to parts of the payload that happen to be encoded at low frequencies. In contrast, Focus's bit error rate is much smoother across the whole payload. Most importantly, errors are concentrated towards the end of the payload, whereas the first half of the payload suffers relatively few errors that can be corrected with forward error correction. This is an effect of Focus's encoding of data in increasingly higher frequencies (§4.1), which ensures that error rates for adjacent bits in the payload are similar. This, in turn, enables effective error correction.

7.2.1 High-end Cameras

The original PixNet paper [29] reports much higher throughput (MBit/s rather than kBit/s) than what we have measured. However, the PixNet authors used high-end DSLR cameras with optical zoom, which provide a much better image quality than smartphone cameras. In contrast, FOCUS targets low-end to mid-range cameras and aims to be robust against their deficiencies, such as low resolution. We nonetheless carry out a simple experiment with a high-end camera as a reader. This experiment serves two purposes: it allows us to understand whether FOCUS is capable of delivering MBit/s throughput on the near-perfect links provided by high-end cameras; and it serves as a validation of our implementation of PixNet.

Experimental setup. We place a Nikon D7100 DSLR camera at a distance of 1.5 m from an LCD. The LCD displays either Focus codes or PixNet codes with a capacity of 66 kbit at a rate of 60 codes/s. We follow the methodology of the PixNet paper [29, see pg. 8] and set the camera's shutter speed to 1/60 s corresponding to an effective frame rate of 30 codes/s.

Results. Both FOCUS and our implementation of PixNet were able to fully decode the data encoded in the respective codes. The throughput for both systems was 2 MBit/s, which is the maximum possible throughput for the given code capacities. Decoding was virtually error free. The results serve as a validation of our PixNet implementation and allow us to conclude that FOCUS can deliver throughput in the order of MBit/s with high-end cameras.

7.3 Throughput under Rate Mismatch

In many scenarios, a transmitter wants to send data to several readers with different capture rates. Instead of adapting its display rate to the slowest reader, Focus's multi-rate streams allow a transmitter to concurrently transmit at different rates. We now evaluate how multi-rate streams alleviate the problems of rate mismatch and frame mixing. In particular, we study the throughput for different ratios of capture rate to display rate.

Experimental setup. To isolate the effects of rate mismatch from the effects of camera quality, we use only the Galaxy S6 with a capture rate of 30 FPS as a reader in this experiment. We vary the display rate from 15 FPS to 60 FPS to study the throughput under various capture/display rate ratios. The reader is placed 75 cm from the display. The transmitter shows a multi-rate stream of Focus codes. One third of the sub-channels of the stream is updated at a rate of 1/4 of the display rate, the next third of sub-channels is updated at a rate of 1/2 of the display rate, and the last third is updated at a rate equal to the display rate.

Results. Fig. 12 shows the experimental results. The x-axis denotes the ratio of capture rate to display rate, i.e., the

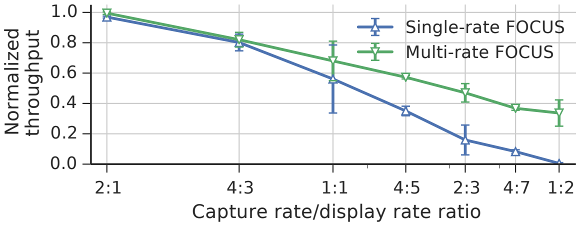


Figure 12: **Impact of mismatch between display and capture rate on the throughput.** FOCUS's multi-rate streams enable readers to receive data even if the display rate is larger than their capture rate.

number of captured frames per displayed code. The y-axis shows the reader's throughput normalized by the sender's data rate. Thus, the y-axis shows which fraction of the transmission could be decoded.

For single rate streams, the highest throughput is reached when the capture/display rate ratio is 2:1, because only then is the reader guaranteed to capture each displayed code at least once without mixing. As the ratio decreases, the throughput decreases, because the reader captures more and more mixed codes. When the ratio is 1:2, the reader cannot decode any data.

In contrast, multi-rate streams enable a transmitter to simultaneously support a wide range of capture/display rate ratios. Even though throughput decreases for readers that capture less than two frames for every displayed frame, the decrease is much smoother for multi-rate codes: if the capture rate is equal to the display rate, the reader can nonetheless decode 66% of the transmitted data; and even if the display rate is twice as high as the capture rate, the reader can decode 33% of the data, albeit every captured frame being mixed. The reader can partially decode the mixed codes because the data on some sub-channels stays constant over multiple frames, as described in §4.3. We conclude that FOCUS multi-rate streams enable a transmitter to concurrently support readers with a range of capture rates.

7.4 Goodput of FOCUS and RDCode

We now study the *goodput* of FOCUS code streams by continuously broadcasting a file and measuring the delay until complete file reception. Note that goodput may differ substantially from throughput on uni-directional screen/camera links if the reader repeatedly misses a specific part of the transmission without the means of explicitly requesting the missing part to be retransmitted. Therefore this section provides a necessary complement to the throughput evaluation. We compare the goodput of FOCUS to RDCode, a recently proposed barcode design for screen/camera links [39]. It is a useful comparison reference, because it is specifically designed for robustness—a goal we also share in the design of FOCUS. The authors of RDCode have provided us with a copy of the source code.

Experimental setup. We transmit from a 3rd generation Apple iPad and receive on the Galaxy S6. The transmitter displays an infinite stream of either RDCode codes or FOCUS codes, which contain a 50.3 KB file. We vary the distance of the S6 to the iPad. We repeat each experiment run five times and show the mean goodput and its standard deviation.

Results. Fig. 13 shows the goodput (file size divided by total transfer time) as a function of the distance between the transmitter and the reader. For distances below 31 cm, we observe a goodput of around 17 KB/s for RDCode, and around 23 KB/s for FOCUS; for larger distances, the goodput of RDCode sharply falls off and the reader is unable to retrieve the file completely. In contrast, FOCUS can maintain the goodput of 23 KB/s for distances up to 50 cm; only beyond this does the goodput gradually decline until it reaches zero at a distance of 87 cm.

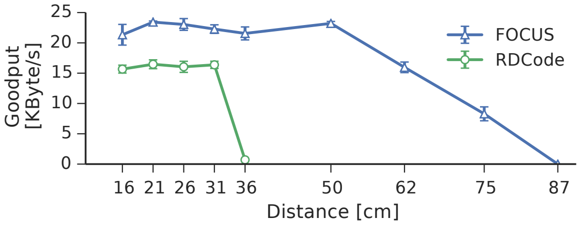


Figure 13: **Goodput of Focus and RDCode for varying distance between transmitter and reader.** FOCUS provides superior goodput and range.

FOCUS consistently outperforms RDCode over all distances in our experiment. While the increase in goodput is moderate (36%), FOCUS supports a significantly larger communication range: its goodput at 62 cm is similar to RDCode's goodput at 31 cm. Even at 75 cm, transferring a 50 KB file only takes 5 s. Examining the implementation internals it seems that RDCode's locator detection algorithm begins to fail at longer distances. We conclude that FOCUS provides improved goodput and communication range over RDCode.

7.5 Embedded FOCUS Streams

For demonstration purposes, we embedded a FOCUS stream into a music video. Fig. 14 shows a frame of the video. We play the music video on a laptop and measure the throughput at the reader, which a user holds in their hand. The embedded stream has a data rate of 30.7 KB/s. In our experiments, we measure the reader's throughput to be 28.6 KB/s. We draw two conclusions from this simple experiment: first, minor hand movements, which are inevitable when the reader is hand-held rather than placed on a tripod, do not significantly degrade throughput. Second, the frequent changes in lighting due to scene changes in the video do not significantly degrade throughput either.

8. APPLICATIONS

FOCUS codes are characterized by their robustness and high capacity. FOCUS codes can be read over long distances and their decoding performance scales with the reader's channel conditions. We now briefly discuss some applications which we expect to benefit most from FOCUS.

We see immediate benefits in scenarios such as logistics, warehouses, and supply chain management. FOCUS codes store more data than traditional visual codes. For example, a QR code can store up to 3 KB, whereas we have successfully used FOCUS codes encoding more than 8 KB. This improved capacity is especially useful to store information that needs to be shared across organizational boundaries. For example, hazardous materials in the supply chain may be labeled with detailed information about their origin and destination, instructions for handling and storing, etc. By embedding all this information in a code, rather than requiring network access and lookup in a cross-organizational database, the

Figure 14: **Music video with an embedded Focus stream.** A recent smartphone can decode data at a rate of 28.6 KB/s from the stream.

information is robustly self-contained and instantly available. Furthermore, extra capacity allows cryptographic signing of data. This enables a reader to verify the authenticity and integrity of a code's information, and renders attacks like code-based phishing ineffective [37, 41].

In processes that require automated identification, such as luggage routing, Focus's improved robustness can reduce the need for costly human intervention by ameliorating decoding failures. Focus codes can also be read over long distances, which is particularly beneficial for scenarios such as warehouse inventory management [24]. Today, long distance reading is only possible with laser-based scanners, which can scan only one code at a time. In contrast, a camera can read multiple Focus codes at once.

Visual codes and markers are also widely used in augmented reality scenarios for pose estimation or to attach digital data to physical objects [1, 18]. With Focus codes, these functions can be combined: data encoded in higher frequencies could, for example, describe different modes of interaction with an object or information to be displayed to users [23], whereas data encoded in the lower frequencies can be used for pose estimation and relative positioning.

Screen/camera links may be used to broadcast data to large audiences, e.g., at sports events, where WiFi and cell networks may be underprovisioned, for broadcasting to many users. Screen/camera links also have the salient security property that they require both nodes on a link to be physically present. In contrast, even short-range RF communication, such as NFC, can be tampered with at a distance using directional antennas [7, 25]. Furthermore, jamming screen/camera links requires the physical presence and visibility of the jammer, thereby raising the bar for such an attack. Alternatively, digital communication through visual codes embedded in a video stream (e.g., TV or web stream), can be used to provide enhancements to the media such as subtitles, without a separate file or data stream.

9. DISCUSSION AND FUTURE WORK

We briefly discuss some of the requirements that Focus makes on transmitters and readers and contrast them with the requirements of existing visual codes. We also describe related prospects for future work.

In contrast to codes that use spatial coding, such as Strata, QR, or DataMatrix, Focus codes represent data in the frequency domain. Therefore, encoding and decoding involve the computation of a Fourier transform. While 2D Fourier transforms can be computationally complex (the running time is $O(n^2 \log n)$), Focus usually operates on small images (e.g., 512×512 pixels), and an implementation can make use of highly optimized libraries for computing Fourier transforms. As such, we do not consider the Fourier transform's overhead to be impractical.

Some visual codes can be represented using black and white only, whereas Focus codes leverage grayscales. Our experience has shown though that Focus codes can be displayed and decoded from devices that support only a limited number of grayscales. For example, Focus codes printed with regular laser printers consist of very small black dots rather than actual grayscales, but can be decoded nonetheless (§6.3). Similarly, we confirmed that Focus codes can be decoded from eInk displays that support only sixteen grayscales; in simulations, we found that Focus codes can even be decoded from black-and-white displays that use dithering to approximate grayscales. In summary, Focus codes can be used over a wide range of display media (LCD, LED, paper, eInk), but a more detailed study of the impact of the characteristics of display media may be of interest in the future.

Focus requires high localization accuracy for codes in a captured photo. As explained in §5.2, low-accuracy localization would cause synchronization errors and lead to inter-symbol interference [29, 33]. However, there are well-established and computationally cheap techniques for sub-pixel localization [4] that we exploit. We are currently exploring alternative designs that use less space for markers while providing comparable localization accuracy.

Finally, modern smartphone cameras can capture several images of a scene in rapid succession. There is a large body of research on how such bursts of images may be used, e.g., to undo the effects of blur [3] or enhance the resolution of the captured images [15]. Understanding whether bursts of photos can improve the decoding performance and data capacity of Focus codes is another direction for future work.

10. CONCLUSION

We presented Focus, a visual code design that is robust to the most common impairments of visual channels. The amount of data that a reader can decode from a Focus code scales automatically with its channel conditions. Our experiments demonstrated the suitability of Focus for smart devices with poor cameras and limited computational capacity. We also showed that Focus significantly advances the state of the art: it improves on the read distance of Strata by a factor 3×, on the throughput of PixNet by at least 2×, and on the communication range of RDCode by 2×.

The key to Focus's performance lies in its encoding of data in many independent sub-channels in the frequency domain. The sub-channels differ from each other in their robustness to channel impairments, and even a reader with a poor channel can decode data from the more robust sub-channels of a code. We conclude that frequency-based visual codes need not be limited to readers with high-end cameras, and that they provide superior capacity and robustness over existing designs.

Acknowledgments

We would like to thank our shepherd Xia Zhou and the anonymous reviewers for their insightful comments. We thank Peter Nyström of IAR Systems for his help with the ARM NEON optimizations. We thank Anran Wang for providing us with a copy of RDCode.

11. REFERENCES

[1] F. Ababsa and M. Mallem. Robust Camera Pose Estimation Using 2D Fiducials Tracking for Real-time Augmented Reality Systems. In *Proc. of VRCAI '04*. ACM, 2004.

[2] A. Ashok, S. Jain, M. Gruteser, N. Mandayam, W. Yuan, and K. Dana. Capacity of Pervasive Camera Based Communication Under Perspective Distortions. In *Proc. of PerCom '14*. IEEE, March 2014.

[3] A. C. S. Atsushi Ito, A. Veeraraghavan, and R. G. Baraniuk. Blurburst: Removing blur due to camera shake using multiple images. *ACM Transactions on Graphics*, to appear.

[4] C. B. Bose and I. Amir. Design of Fiducials for Accurate Registration Using Machine Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Dec. 1990.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A Digital Fountain Approach to Reliable Distribution of Bulk Data. *SIGCOMM Computer Communincation Review*, 28(4), Oct. 1998.

[7] T. P. Diakos, J. A. Briffa, T. W. C. Brown, and S. Wesemeyer. Eavesdropping Near-Field Contactless Payments: a Quantitative Analysis. *The Journal of Engineering*, January 2013.

[8] European Committee for Standardization. *Automatic Identification and Data Capture Techniques – QR Code 2005 Bar Code Symbology Specification: ISO/IEC 18004-2006*. 2006.

[9] M. Frigo. A Fast Fourier Transform Compiler. In *Proc. of PLDI '99*. ACM, 1999.

[10] T. Hao, R. Zhou, and G. Xing. Cobra: Color Barcode Streaming For Smartphone Systems. In *Proc. of MobiSys*. ACM, 2012.

[11] F. Hermans, L. McNamara, and T. Voigt. Demo: Scalable Visual Codes for Embedding Digital Data in the Physical World. In *Proc. of SenSys '15*. ACM, 2015.

[12] S. Hranilovic and F. Kschischang. A Pixelated MIMO Wireless Optical Communication System. *IEEE Journal of Selected Topics in Quantum Electronics*, 12(4), 2006.

[13] W. Hu, H. Gu, and Q. Pu. LightSync: Unsynchronized Visual Communication over Screen-camera Links. In *Proc. of MobiCom '13*. ACM, 2013.

[14] W. Hu, J. Mao, Z. Huang, Y. Xue, J. She, K. Bian, and G. Shen. Strata: Layered Coding for Scalable Visual Communication. In *Proc. of MobiCom '14*. ACM, 2014.

[15] Y. Huang, W. Wang, and L. Wang. Bidirectional recurrent convolutional networks for multi-frame super-resolution. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015.

[16] ISO. *Information technology—Automatic Identification and Data Capture Techniques—Data Matrix Bar Code Symbology Specification*. Number ISO/IEC 16022:2006. Geneva, Switzerland, 2006.

[17] ISO. *Information Technology – Automatic Identification and Data Capture Techniques – Aztec Code Bar Code Symbology Specification*. Number ISO/IEC 24778:2008. Geneva, Switzerland, 2008.

[18] T.-W. Kan, C.-H. Teng, and W.-S. Chou. Applying QR Code in Augmented Reality Applications. In *Proc. of VRCAI '09*. ACM, 2009.

[19] V. Lepetit and P. Fua. Monocular Model-based 3D Tracking of Rigid Objects. *Foundations and Trends in Computer Graphics and Vision*, 1(1), Jan. 2005.

[20] A. Lerner, A. Saxena, K. Ouimet, B. Turley, A. Vance, T. Kohno, and F. Roesner. Analyzing the Use of Quick Response Codes in the Wild. In *Proc. of MobiSys*. ACM, 2015.

[21] T. Li, C. An, X. Xiao, A. T. Campbell, and X. Zhou. Real-Time Screen-Camera Communication Behind Any Scene. In *Proc. of MobiSys '15*. ACM, 2015.

[22] R. LiKamWa, D. Ramirez, and J. Holloway. Styrofoam: A Tightly Packed Coding Scheme for Camera-based Visible Light Communication. In *Proc. of MobiCom Workshop on Visible Light Communication Systems*. ACM, 2014.

[23] S. Mayer and G. Soros. User Interface Beaming – Seamless Interaction with Smart Things Using Personal Wearable Computers. In *Proc of. Wearable and Implantable Body Sensor Networks Workshops (BSN Workshops)*, June 2014.

[24] Motorala Inc. The Next-Generation Warehouse — Long Range Scanning And Emergence of 2D Bar Codes. Technical report, 2011.

[25] C. Mulliner. Vulnerability Analysis and Attacks on NFC-Enabled Mobile Phones. In *Proc. of ARES '09*, pages 695–700, March 2009.

[26] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab. *Signals and Systems*. Prentice-Hall, Inc., NJ, USA, 1996.

[27] D. Parikh and G. Jancke. Localization and Segmentation of a 2D High Capacity Color Barcode. In *Proc. of Workshop on Applications of Computer Vision*. IEEE, 2008.

[28] S. D. Perli. MIT Masters Thesis - PixNet: Designing Interference-free Wireless Links Using LCD-camera Pairs, 2010.

[29] S. D. Perli, N. Ahmed, and D. Katabi. PixNet: Interference-free Wireless Links Using LCD-camera Pairs. In *Proc. of MobiCom '10*. ACM, 2010.

[30] T. Pollet, M. Van Bladel, and M. Moeneclaey. BER Sensitivity of OFDM Systems to Carrier Frequency Offset and Wiener Phase Noise. *IEEE Transactions on Communications*, 43(2/3/4), Feb 1995.

[31] Y. Rahmatallah and S. Mohan. Peak-To-Average Power Ratio Reduction in OFDM Systems: A Survey And Taxonomy. *IEEE Communications Surveys Tutorials*, 15(4):1567–1592, April 2013.

[32] I. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *Journal of the Society of Industrial and Applied Mathematics*, 8(2), June 1960.

[33] T. Schmidl and D. Cox. Robust Frequency and Timing Synchronization for OFDM. *IEEE Transactions on Communications*, 45(12), Dec 1997.

[34] S. Shi, L. Chen, W. Hu, and M. Gruteser. Reading Between Lines: High-rate, Non-intrusive Visual Codes

Within Regular Videos via ImplicitCode. In *Proc. of UbiComp '15*. ACM, 2015.

[35] A. Shokrollahi. Raptor Codes. *IEEE/ACM Transactions on Networking*, 14(SI), June 2006.

[36] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. CL Engineering, 2007.

[37] T. Vidas, E. Owusu, S. Wang, C. Zeng, L. Cranor, and N. Christin. QRishing: The Susceptibility of Smartphone Users to QR Code Phishing Attacks. In *Financial Cryptography and Data Security*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013.

[38] A. Wang, Z. Li, C. Peng, G. Shen, G. Fang, and B. Zeng. InFrame++: Achieve Simultaneous Screen-Human Viewing and Hidden Screen-Camera Communication. In *Proc. of MobiSys '15*. ACM, 2015.

[39] A. Wang, S. Ma, C. Hu, J. Huai, C. Peng, and G. Shen. Enhancing Reliability to Boost the Throughput over Screen-Camera Links. In *Proc. of MobiCom '14*. ACM, 2014.

[40] Q. Wang, M. Zhou, K. Ren, T. Lei, J. Li, and Z. Wang. Rain Bar: Robust Application-Driven Visual Communication Using Color Barcodes. In *Proc. of ICDCS '15*, June 2015.

[41] H. Yao and D. Shin. Towards Preventing QR Code Based Attacks on Android Phone Using Security Warnings. In *Proc. of ASIA CCS '13*. ACM, 2013.