Pervasive and Mobile Computing 16 (2015) 239–250

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Capacity of screen–camera communications under perspective distortions

Ashwin Ashok*, Shubham Jain, Marco Gruteser, Narayan Mandayam, Wenjia Yuan, Kristin Dana

Wireless Information Network Laboratory (WINLAB), Rutgers University, 671 Rt 1 South, North Brunswick, NJ, 08902, USA

ARTICLE INFO

Article history:
Available online 15 November 2014

Keywords:
Screen–camera communication
Visual MIMO
Camera communication
Perspective distortion
Capacity

ABSTRACT

Cameras are ubiquitous and increasingly being used not just for capturing images but also for communicating information. For example, the pervasive QR codes can be viewed as communicating a short code to camera-equipped sensors. Such communications could be particularly attractive in pervasive camera based applications, where such camera communications can reuse the existing camera hardware and also leverage from the large pixel array structure for high data-rate communication. While several prototypes have been constructed, the fundamental capacity limits of the screen–camera communication channel in all but the simplest scenarios remains unknown. The visual medium differs from RF in that the information capacity of this channel largely depends on the perspective distortions while multipath becomes negligible. In this paper, we create a model of this communication system to allow predicting the capacity based on receiver perspective (distance and angle to the transmitter). We calibrate and validate this model through lab experiments wherein information is transmitted from a screen and received with a tablet camera. Our capacity estimates indicate that tens of Mbps is possible using a smartphone camera even when the short code on the screen images onto only 15% of the camera frame. Our estimates also indicate that there is room for at least 2.5× improvement in throughput of existing screen–camera communication prototypes.

1. Introduction

The pervasive use of cameras has led to not only a diverse set of camera-based sensing applications but also to novel opportunities to use cameras to communicate information [1]. Recent efforts to standardize camera communications [2] attests to the importance of using camera for communications. Camera based communication is characterized by highly directional transmission and reception along with low-multipath interference rendering it virtually interference-free. Thus, it is particularly attractive for dense congested environments where RF communication data rates are largely limited due to interference, for security applications where the directional transmissions lead to lower probability of interception or observability of signals, or for situations where the high directionality leads to improved localization of the transmitters. Camera based communication can leverage existing cameras for communicating with the ubiquitous light emitting devices. Information could be transmitted from TVs, monitors, billboards, and even projector screens. We believe, therefore, that camera-based communications can be an attractive alternative or supplement to RF wireless based communication.

* Corresponding author. Tel.: +1 8489324381.
E-mail address: aashok@winlab.rutgers.edu (A. Ashok).

http://dx.doi.org/10.1016/j.pmcj.2014.11.003
1574-1192/

Today, cameras are frequently used to read QR-codes, which can be considered as a form of visual communication wherein the camera acts as a receiver. The ubiquitous use of QR codes motivates building novel camera communication applications, where pervasive display screens could be modulated to send time-varying QR codes to be decoded by video cameras. The large pixel array elements of the screen and camera can be leveraged to send high volume of data through short time-varying 2D barcodes. For example, a user could point a camera to a desktop PC or a smartphone screen displaying the time-varying code to download a file. Recent research has further explored this direction by designing prototypes wherein time-varying 2D barcodes can be transmitted from desktop monitors [3,4] and smartphone screens [5] to a camera receiver. While these works have designed and measured the information capacity for a specific point solution in this space, how much room for improvement exists in these solutions or if there is any bound on performance still remains unclear.

To the best of our knowledge, only few projects have begun to investigate information capacity limits of camera communication using time-varying 2D barcodes. Hranilovic et al. [4] analyzed the capacity and prototyped a screen–camera system where a CCD camera was placed at a fixed distance from a laptop screen. The model does not account for the interference between pixels and the dependence on perspective. The model in [1] can be considered as a simplified case of screen–camera channel where the transmitter and receiver always remain aligned, while ignoring the quantization effects of real camera receivers.

In this paper, we develop a model for the information capacity of screen–camera communication that accounts for perspective dependent (position and orientation) distortions that dominate this channel. The model incorporates projection theory from the computer vision domain into a Shannon capacity formulation. Specifically, our contributions in this paper are:

- A screen–camera channel model that accounts for perspective distortions and realities of camera receivers such as quantization limitations, and blur.
- Experimental calibration and validation of the model through extensive lab measurements using a screen transmitter and a tablet camera receiver. The studied parameters include distance, angle, the granularity or block-size of the code (number of pixels per transmit bit), and motion blur.
- Estimation and validation of capacity for screen–camera communication by measuring channel and signal quality metrics, such as bandwidth and signal-to-interference-noise ratio, and substituting into the derived analytical capacity expression.
- A comparison of capacity estimate with throughput of existing screen–camera communication prototypes.

2. Related work

Camera based communication is an example of visual MIMO communication [1] where camera is used as a receiver for information transmitted from arrays of light emitting elements. In our earlier work in [1] capacity of a camera channel was estimated by treating the transmitter light emitting array and the camera perfectly aligned. The channel is considered as an analog communications channel where the signal at the receiver is the sampled photocurrents from each image pixel, and do not take into account the quantization limitations in the camera.

The LCD screen–camera channel capacity estimates [4] were based on a water-filling algorithm assuming the camera channel can be equalized to encounter the effects of spatial distortions. But the model and the prototype were designed for a fixed distance of 2 m between the screen and camera and did not study the effects of perspective on the estimated capacity and throughputs achieved. Perspective distortion has been studied by the imaging community previously [6,7], but the fact that the camera is a part of a wireless communication channel (captured object is the light source itself) presents a new domain of challenge for applying imaging models in analyzing communication channels.

The advent of high-resolution cameras in mobile devices has spurred interest in using cameras for communication to retrieve information from screens [5,8,9,3]. These applications use specific receiver processing schemes to combat visual distortions. PixNet [3] proposes to use OFDM modulation to combat the effect of perspective distortion on images by inverse filtering on the estimated channel, and using forward error correction. COBRA [5] proposes to leverage from encoding on the color channels to achieve throughput gains for smartphone screen–camera communication, but at very short distances (22 cm). The fact that several prototypes have been constructed reveals that screen–camera communication is gaining large momentum.

3. Background on camera communication

A camera channel is analogous to a RF MIMO channel where each pixel element of the camera acts as a receiving antenna and the light emitting elements as the transmit antennas. In RF MIMO, the signal quality at each receive antenna element is a function of the path-loss in the channel, multipath fading, and the interference from other transmit antennas—also called co-channel interference [10]. A camera channel has negligible multipath fading, but experiences path-loss in light energy, and interference (of light energy) from other light emitting elements, which manifest as visual distortions on the output of a camera; that is, the image. These distortions are primarily a derivative of the camera imaging process and can be modeled (deterministically) using classical camera imaging theory.

The signal quality at the camera receiver is also influenced by noise in the channel. Noise in camera systems manifests as spurious electric signal in the form of *current* on each camera pixel. Noise current is generated due to the photons from, environment lighting (includes ambient lighting) and from the transmitter and receiver imaging circuitry [11]. Noise current in a pixel is usually considered signal independent when the ambient lighting is sufficiently high compared to the transmit signal; for example, in office rooms or outdoors [12]. At the output of a camera, the noise current in each camera pixel is a quantized quantity and manifests as fluctuations in the intensity (digital value of the sensor output) of that pixel; the noise energy accumulated in each pixel can be quantified using the mean value of variance in the pixel intensity. As in prior works that modeled optical channels [12,13], in this paper, we consider that the noise in a camera pixel is primarily from the background, and follows a AWGN characteristic (quantified through the AWGN noise-variance σ_n^2), and is uniform over the image sensor (photoreceptor).

Considering the deterministic nature of perspective distortions and the AWGN characteristic of the camera channel, capacity (measured in bits/sec) of camera based communication can be expressed using Shannon Capacity formula as,

$$C = W_{fps}(W_s \log_2(1 + \text{SINR})) \tag{1}$$

where *SINR* represents the signal-to-interference-noise ratio per pixel, W_{fps} is the camera-frame rate or the receiver sampling rate in frames-per-second. W_s is the spatial-bandwidth, which denotes the number of information carrying pixels per camera image frame. The spatial bandwidth is equivalent to the number of orthogonal or parallel channels in a MIMO system.

Throughout this paper, we use the terms *screen transmitter* and *screen* interchangeably, and the term *image* to refer to the camera sampled image.

4. Screen–camera channel

In screen–camera communication, information is modulated in the light intensity of the pixel elements of a screen transmitter that are received and decoded from the camera image pixel intensity at the receiver. The pixel intensity in a camera image is a digital quantity[1] that is proportional to the amount of photon current generated on the pixel from the light energy accumulated over its area (the smaller the pixel area the lesser light intensity it accumulates). When the light emitting screen pixel is at the focus of the camera lens all the light rays from the screen pixel are focused onto a camera pixel and thus incurring no loss of energy on the pixel. When the screen pixel is perturbed (in position and/or orientation) from the focus of the camera or incurs path-loss in energy due to the finite aperture size of the camera lens, not all light rays converge on the camera pixel resulting in reduced accumulated energy and hence a lower pixel intensity value. The loss in the received light intensity on a camera pixel results in the visual deformation in size or shape of the imaged screen pixel; an effect that is termed as perspective distortion.

Loss in signal energy on a pixel is also attributed to the noise in that pixel. As discussed earlier, noise in a camera pixel is primarily due to spurious photons (that do not belong to the transmitter) from the environment, which can be modeled as signal independent and AWGN. Noise from the transmitter and the camera imaging circuit are dependent on the generated signal (and that is transmitted), and thus depend on the transmitter and receiver specifications. However, unlike environment noise, this signal dependent noise can be estimated using one-time calibration mechanisms; camera noise modeling has been well studied in computer vision and solid-state electronics (CMOS) design literature. We reserve the discussions on effect of signal dependent noise on throughput of camera communications for future work.

4.1. Perspective distortions

Distortions that depend on the perspective of the camera are caused due to the nature of the camera imaging mechanism and manifest as deformation in size and shape of the captured object (the light emitting screen pixel) on the image, resulting in visual compression or magnification of the object's projection on the image. When the screen is at an out-of-focus distance from the camera lens (or at an oblique angle), these distortions become prominent and lead to interference between adjacent screen pixels on the camera image, what we term as inter-pixel interference or IPI. The combined effect of background noise and IPI degrades the received signal quality and hence reduces information capacity in camera channels.

For example, let us consider that blocks of pixels on a screen are illuminated by a chessboard pattern and imaged by a camera as shown in Fig. 1. We can observe that perspective distortions cause the screen pixels to deform in size when the screen is not at the focus of the camera, and in shape when it is not frontally aligned (viewed at an angle) with the camera.

Perspective scaling. If the screen pixel was at the focus, and assuming the screen and camera have the same resolution, its image on the camera should occupy the same area as one pixel. But in reality, the light rays from the screen pixel may not end exactly on camera pixel boundaries and there is some area surrounding it that accumulates interference. This area of misalignment and the geometry of the imaged screen pixel will be perspective dependent and accounts for distortion due to perspective scaling of the pixel area.

[1] Most cameras have 8 bit monochromatic depth (on each color channel) where the values span 0 (dark)-to-255 (bright).

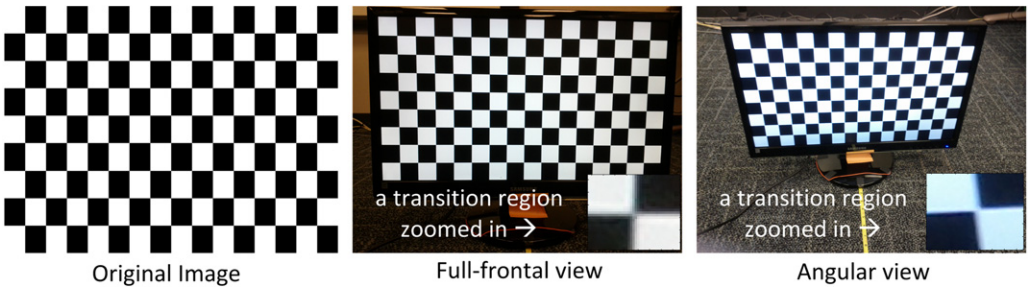


Fig. 1. Illustration of perspective distortion in screen–camera channel. Imaged screen pixels are blurry, and reduced in size in full-frontal view and also in shape in angular view.

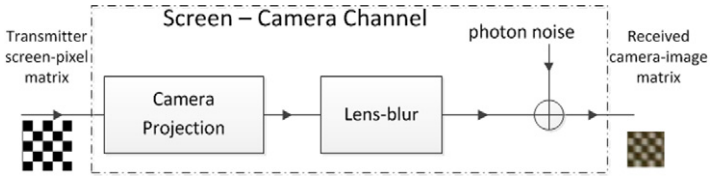


Fig. 2. Screen–camera channel model.

Lens-blur. We can also observe from Fig. 1 that the imaged screen pixels are blurry, especially at the transition regions between white and black blocks. This blur effect is attributed to the camera lens and more formally termed as lens-blur. This blur effect is typically modeled in camera imaging theory using the point-spread function (PSF) [14], which represents the response of an imaging system to a point source. In the screen–camera channel this translates to distorting the pixels at the transition regions between brighter (high intensity) and darker (low intensity) pixels, and leads to interference (IPI) between neighboring pixels, as seen in Fig. 1. Since the area and the maximum energy that can be sampled in each camera pixel is finite, IPI leads to an effective reduction in signal energy per pixel.

5. Modeling perspective distortion factor

In this paper, we model the perspective distortions in the screen–camera channel as a composite effect of signal energy reduction due to perspective scaling of pixel area owing to camera projection, signal energy reduction due to lens-blur, and background photon noise, as shown in Fig. 2. In this regard, we consider that the signal energy on each pixel is weighted by an average perspective distortion factor α, that represents the effective area scaling (down) due to perspective and lens-blur in the camera imaging process, while the rest of the light-energy on the pixel is from ambient photon noise. We define this factor such that it takes values in $0 \leq \alpha \leq 1$, where $\alpha = 1$ indicates that the screen pixel is at the focus of the camera and also incurs no signal reduction due to lens-blur, and $\alpha = 0$ indicates that no part of the screen-pixel gets imaged on the camera pixel.

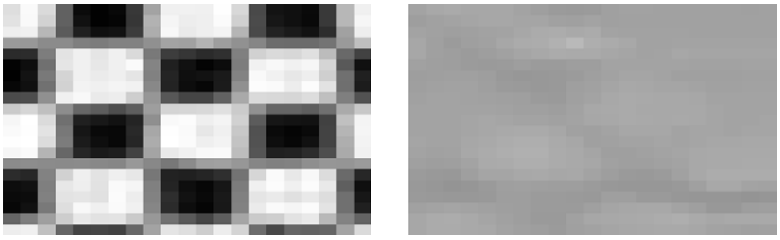
Perspective scaling. Let α_p represent the perspective scaling of the area of an imaged screen pixel when perturbed from camera focus. We modeled this perspective scaling factor and derived a general expression for α_p in [15] using camera projection theory [16], that uses the camera projection matrix which maps the location of the screen pixels from the world coordinate frame to the camera coordinate system. In the simplest case, where the screen and camera are perfectly aligned at distance d, this factor can be expressed as,

$$\alpha_p = \left(\frac{f_{cam} s_t}{s_{cam} d}\right)^2 \tag{2}$$

where f_{cam}, s_t are the focal length of the camera and side-length of the screen pixel, respectively, and s_{cam} is the side length of the camera pixel. We can observe from Eq. (2) that, $\alpha_p = 1$ when the camera is at the focus ($d = f_{cam}$) and if $s_{cam} = s_t$. However, in reality, the physical size of a screen and camera pixel may not be the same. In our system, we assume that the focal point is at a distance $d_f = \frac{f_{cam} s_t}{s_{cam}}$ to the screen; which we term as *focal-distance*.

Lens-blur. As discussed earlier, lens-blur causes the signal energy to leak outside the area of a single pixel. Camera lens-blur, characterized by the PSF, can be approximately modeled as a 2D Gaussian function [14,17], where the amount of spread in area is quantified using its variance σ^2_{blur} (a large variance indicates more blur[2]). In our model we account for lens-blur

[2] For an ideal pin-hole camera energy spread over a pixel would be uniform and hence σ^2_{blur} is infinitesimally small.



(a) Handheld. (b) Motion.

Fig. 3. Illustration of motion blur on images of a screen displaying a chessboard pattern, taken by a hand-held camera (a) and when camera is in motion (b).

distortion using the factor $\alpha_b = (2\sigma_{blur})^2$, to account for the spread in area over two dimensions of the square pixel. If s_{cam} is the side length of a camera pixel, then the effective signal energy on that pixel will be proportional to $s_{cam}^2 \frac{1}{1+\alpha_b}$. We treat this signal energy reduction is proportional to this reduced pixel area over which the signal accumulates.

In this regard, we consider α, an average distortion in each pixel of the camera image to quantify perspective distortion. We express α as the effective pixel area reduction due to perspective scaling factor α_p on the reduced pixel area due to lens-distortion $\alpha_b = 4\sigma_{blur}^2$, as

$$\alpha = \alpha_p \times \frac{1}{(1+\alpha_b)}. \tag{3}$$

In reality, the physical size of the screen pixel may not exactly be matched with that of the camera image sensor. The screen and camera image sensor pixels may not be of the exact same size and even if so they may not be exactly similar rectangles and some micro-deviations can still persist which can cause misalignments in the imaged pixels. This can cause an imaged screen pixel not to align with a camera pixel, even if the screen pixel were at the camera focus. Such misalignments will cause a deviation in the distortion factor for each pixel as the perspective changes. However, such deviations can be assumed to be negligible when compared to the distortions due to perspective scaling. By considering an average distortion factor over the camera image such micro-deviations will be almost negligible.

5.1. Discussion on motion blur

Before we step into modeling the signal quality metrics of a screen–camera channel we discuss the effect of motion on screen–camera communication. Screen–camera communication applications would typically involve some degree of motion, for example, when the camera is hand-held, or when the camera or screen is on a moving object such as a vehicle. Motion due to hand-shakes or lateral movements can cause dynamic change in perspective between the screen and the camera. In such cases, one can assume some vibrations on the pixels, especially when the camera is not stable, where the pixels seem to interfere with each other, eventually causing a blurry visual effect on the image; formally known as motion-blur in computer vision [14].

Motion-blur primarily arises due to movement within or between camera frames. Smartphone cameras are usually hand-held and vibrations caused due to hand motion can cause motion blur but are usually much less than those when the screen or camera is in motion. Cameras equipped in vehicles may suffer from more blur compared to hand-held scenarios as camera sampling may be too slow when compared the speed of motion. Fig. 3 shows an example of camera snapshots of a screen imaged when camera is (a) hand-held, and (b) in motion. We can observe from these snapshots the distortions due to motion blur leading to inter-pixel interference.

Cameras today are equipped with very effective *motion compensation* capability which compensate motion blur through a filtering mechanism called *de-blurring*. De-blurring [18,19] is a technique that is commonly used to mitigate the effect of blur on the image by applying a filter that inverts the effect of blur on the image. The quality of the de-blurred image will largely depend on the effectiveness of the de-blurring filter as well as the amount of induced motion/vibration on the pixels. Imperfections in the de-blurring process can also lead to signal quality reduction compared to an ideal (static screen and camera) scenario. If the motion is fast then the camera may not be able to expose to the entire screen pixel and hence causing the signal energy to spread over many pixels and result in a more blurry image as shown in Fig. 3(b).

This section essentially provides, to the reader, a discussion on motion-blur, how it is related to the inter-pixel interference in screen–camera communication, and how it can possibly be mitigated, through de-blurring. The effects of de-blurring will depend on what algorithm is chosen. In the interest of this paper, we reserved such considerations for future work. Since we study the bounds on capacity we considered the fixed case scenario as the benchmark scenario for our model, which essentially would be best-case scenario in reality. When modeled motion blur would factor into the α_b parameter.

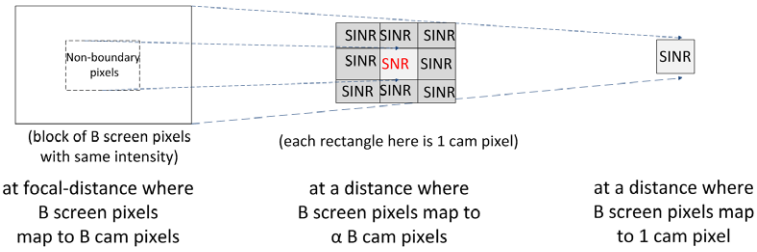


Fig. 4. Illustration of interference between pixel-blocks due to perspective distortion for SINR computation.

6. Signal-to-interference noise ratio in screen–camera channel

We quantify the quality of the signal at the camera receiver in the screen–camera channel using the average SINR per pixel,

$$\text{SINR}_\alpha = \frac{\alpha P_{avg}^2}{(1-\alpha)P_{avg}^2 + \sigma_n^2} \tag{4}$$

where, P_{avg} denotes the average transmit pixel intensity. For example, a screen–camera system using black (digital value 0) and white (digital value 255) intensities for transmission will have $P_{avg} = 127.5$. By using the digital value of the average signal P_{avg}, instead of its analog equivalent (pixel photon-current squared), our model accounts for the quantization limitations in cameras. The $1-\alpha$ term in Eq. (4) quantifies the fraction of the pixel area affected by interference.

6.1. Pixel blocks

A small value of α indicates that more screen pixels interfere on one camera pixel. In reality, screen pixels are very closely spaced (fraction of a mm), and so, IPI will be inevitable even at short distances resulting in low SINRs. A potential solution is to leverage the MIMO structure of the screen–camera channel, by grouping multiple screen pixels in a block, such as a 2D barcode, to transmit with same intensity, and combine such pixels from the camera image to improve SINR. This technique, in principle, is similar to diversity combining used in RF MIMO. Pixel-blocks merely represent that a group of antennas are used to transmit the same intensity, to improve the SINR at the receiver. By using pixel blocks, we draw analogies of the screen–camera channel to an equivalent MIMO system. This is different from considering multiple-level modulation or coding to improve communication throughput. In this paper we are primarily interested in determining the bounds on the information capacity which by definition is independent of the type of modulation or coding used.

Pixel blocks are effective in reducing the impact of misalignments, and lens-blur, as these effects become smaller as one block covers more pixels on the camera and only affect pixels near the boundary as shown in Fig. 4. The SINR can be enhanced by considering averaging the signal energy over such blocks of pixels.

As a convention in our model, we treat a pixel block as a boundary block if it is not all surrounded by blocks with same intensity. Such a structure minimizes the 'interference' for a non-boundary pixel, and is negligible when the camera and screen are static with respect to each other. In this case, even for non-zero blur or pixel misalignment, since the same signal adds-up on the pixel, it enhances signal energy of that pixel; in which case the SINR of that pixel converges to the average-SNR.

In general, the expression for the average SINR per imaged block in a screen–camera channel, using B pixel square blocks of a screen can be given as,

$$\begin{aligned} \text{SINR}_{blk}(\alpha, B) &= \gamma_1 \text{SINR}_\alpha + \gamma_2 \text{SNR}_\alpha \quad \forall \alpha B > 4 \\ &= \text{SINR}_\alpha \quad \forall \alpha B \leq 4 \end{aligned} \tag{5}$$

where SINR_α is from Eq. (4), $\text{SNR}_\alpha = \frac{\alpha P_{avg}}{\sigma_n^2}$, and the coefficients $\gamma_1 = 4(\sqrt{\alpha B} - 1)$ and $\gamma_2 = (\sqrt{\alpha B} - 2)^2$ represent the number of boundary-blocks and non-boundary blocks, respectively. Here, $\min B = 4$ (i.e. 2×2 pixels), and $\alpha B \leq 4$ indicates that each B pixel block projects onto a maximum of 1 camera pixel area while $\alpha B > 4$ indicates that the block projects onto multiple camera pixels.

7. Capacity under perspective distortions

Recalling the capacity expression from Eq. (1), we can express the capacity of screen–camera communication in bits/s as,

$$C_{cam}(\alpha) = \frac{W_{fps}}{2} \alpha \|R_{cam}\| \log_2(1 + \text{SINR}_\alpha) \tag{6}$$

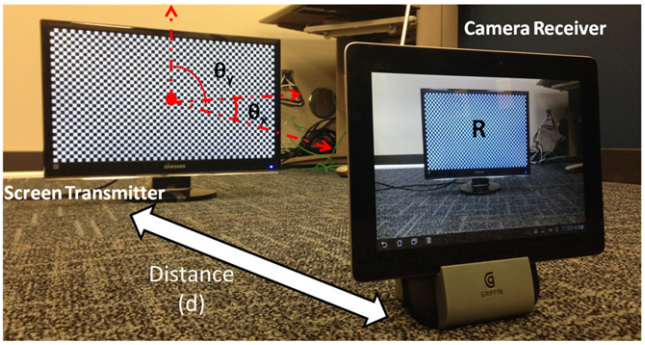


Fig. 5. Experiment setup showing LCD screen displaying black and white blocks of $B = 60 \times 60$ pixels each.

where SINR_α is the signal-to-interference noise ratio from Eq. (4), $\|R_{cam}\|$ denotes resolution of the camera and W_{fps} denotes the frame-rate of the camera in frames-per-second. The camera frame-rate, and hence bandwidth, is halved (following Nyquist sampling theory) to avoid the mixed frames caused by aliasing resulting from the synchronization mismatch between screen updates and the camera sampling. The term $\alpha\|R_{cam}\|$ represents the total number of camera pixels that contain the image of the screen pixels, and directly corresponds to the spatial-bandwidth term W_s in Eq. (1). This is very different from RF MIMO, where, all the receiver antennas can potentially receive the signal, independent of distance between the transmitter and receiver. In a camera receiver, due to its LOS nature, the signal from each transmit element is always limited to a finite number of, but never all, receive elements.

MIMO throughput. The capacity in Eq. (6) represents the upper bound on the total number of bits that can be communicated with negligible error from one screen pixel to a camera pixel. Grouping pixels into blocks improves the SINR and reduces bit errors, but the effective data throughput scales down as the number of parallel channels are reduced. This behavior is similar to the classical multiplexing-diversity tradeoff in RF-MIMO [20]. If $T_{blk}(\alpha, B)$ represents the MIMO capacity or maximum throughput of screen–camera communication for block-size B, at distortion factor α, then

$$T_{blk}(\alpha, B) = \frac{W_{fps}}{k}\left(\frac{\alpha\|R_{cam}\|}{B}\right)\log_2(1 + \text{SINR}_{blk}(\alpha, B)), \tag{7}$$

where $\frac{\alpha\|R_{cam}\|}{B}$ represents the number of parallel channels for multiplexing, and $\text{SINR}_{blk}(\alpha, B)$ is from Eq. (5). In practice, to minimize detection and decoding errors, the camera frame-rate has to be synchronized with the modulation rate of pixel intensities on the screen as well as the refresh rate of the screen (typically 120 Hz). The factor k in Eq. (7) corresponds to the oversampling factor to address the asynchronism between the screen (data) update rate and the camera sampling rate. It implies that a minimum of k temporal samples of the camera pixel are required for reliable decoding. Synchronization of cameras for communication is challenging due to the jittery nature (owing to software limitations and hardware design errors) of the frame-sampling using CMOS sensors that are widely used in mobile devices today.

8. Experimental calibration and validation

In this section we describe the experiments we conducted to validate our screen–camera channel model. The key motive of these experiments was to determine the channel capacity in a real screen–camera channel.

Measured channel capacity. It is a fact that it not possible to measure capacity of any communication channel directly, hence we aim to determine capacity indirectly by substituting the measured SINR, perspective distortion factor α and noise power[3] into the analytical capacity expression derived in (6). Our experiments were aimed at measuring these specific parameters that aid in determining capacity values for an example test channel that we considered. However, we note that these experiments as well as the findings can be applied to a generic camera communications channel—with appropriate specifications of the transmitter and receiver considered. In this paper, we estimate capacity of screen–camera channel by substituting the measured values of SINR_α, perspective distortion factor α, and noise variance σ_n^2 in Eq. (6). The measurement procedure for α, SINR_α are explained in detail in Sections 8.3 and 8.4 respectively.

8.1. General experiment methodology

The experiment setup, as shown in Fig. 5, consisted of a 21.5 inch Samsung LCD screen monitor of resolution $R_s = 1920 \times 1080$ pixels, that served as the screen-transmitter, and a 8 MP camera of a ASUS Transformer Prime tablet (that ran Android

[3] We encourage the reader to refer to [15] for details on the AWGN noise measurement procedure.

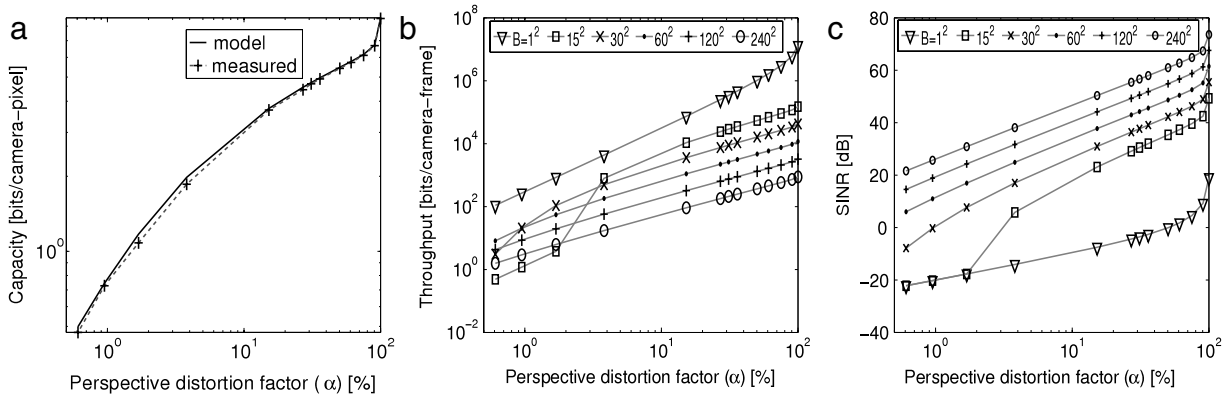


Fig. 6. (a) Capacity in bits/camera pixel ($C_{campixel}(\alpha)$) for different perspective scaling (α) of screen image on camera (b) Throughput in bits/frame vs. α for different blocksizes (1 frame = R_{cam} pixels, $B = 15^2$ means 15×15 pixel block on screen) (c) SINR per block vs. α for different blocksizes B.

OS version 4.1), that served as the camera receiver. The camera was operated at a resolution of $R_{cam} = 1920 \times 1080$ and with no image compression. Exposure setting and white-balancing on the camera were set to auto (default setting in Android devices). All the experiments were conducted under the same environment lighting conditions with the measurements taken indoors in a lab-conference room setting equipped with fluorescent ceiling lighting. We fixed the screen and tablet onto stands so as to ensure the least amount of error in the measurement of distance and angle between the tablet and camera image planes. The raw dataset for our analysis consisted of image snapshots of the screen, displaying a chessboard pattern (blocks of B pixels each), captured by the tablet's camera at resolution of R_{cam} pixels using a standard Android image capture application. The camera parameters were obtained through a well known calibration toolbox [21]. The pixel-intensity of a white block was set to 255 and the black at 25[4] on the screen (the average intensity $P_{avg} = 140$). The image datasets consisted of 100 snapshots of the screen displaying the chessboard pattern, with the ceiling lights ON (an another dataset with lights OFF), at a set of distances, angles, and block-sizes. We changed angle between screen and camera by rotating the screen with respect to the X axis; distortions can be considered symmetrical on X and Y axis.

Table 2 summarizes the list of measured parameters from our experiments, along with the screen and camera specifications.

8.2. Channel capacity

We evaluate capacity in bits per camera pixel as $C_{campixel}(\alpha) = \frac{C_{cam}(\alpha)}{\frac{W_s}{2} \| R_{cam} \|}$.

8.2.1. Capacity vs. perspective distortion factor

We plot the measured capacity in bits/camera-pixels for different perspective distortion factor values in Fig. 6(a) along with the analytical values, and observe a good fit (maximum error margin of 3%) between the two. The distortion factor α on the x-axis is comprehensive of composite distortion due to perspective scaling as well as blur. We can observe that, about 1 bit/camera pixel is achievable even when the screen is perspectively scaled onto only 15% on each dimension ($\alpha = 2\%$) of the camera image. For the LCD screen-tablet camera system we used, this translates to a distance of 2.6 m. At a sampling rate of 30 fps[5] and at a resolution of 1920×1080, a data-rate of 31 Mbps is achievable from an average-sized LCD monitor and a tablet camera. Assuming all parameters are the same, except the size of the screen is doubled, the same data-rate can be achieved at twice the range. Such data-rates are even sufficient for streaming a video.

8.2.2. Throughput with block-size

We plot the screen–camera communication throughput from Eq. (7) in bits-per-frame ($\frac{T_{blk}(\alpha,B)}{kW_{fps}}$) for different values of perspective distortion factors, and block sizes B, in Fig. 6(b). We can observe from Fig. 6(b) that capacity falls of steeply as α becomes smaller for smaller block-sizes; for example, at $B = 15^2$ and 30^2. The trend can be attributed to the low SINR at those perspectives as IPI increases due to the dense arrangement of bits (pixels carrying unique information). A block-size of 1 does not follow this trend as the gain from the capacity scaling due to more number of parallel channels compensates for most of the loss in SINR, however, trading-off with receiver complexity to detect the very low SINR signal.

[4] Due to the screen's residual back-lighting, intensities in [0, 25] range did not cause any change in screen brightness.

[5] Typical frame-rate on smartphone/tablet cameras is 30 fps. iPhone 5S has a 120 fps capability [22].

Table 1
Ratio of capacity over existing prototype's throughput (3× indicates the existing prototype is 1/3rd of capacity).

COBRA	PixNet-C	PixNet	QR-P
4.5×	3×	2.5×	7×

Table 2
Table of screen, camera and measured parameters.

Parameter	Value
Cam pixel side-length s_{cam} (μm)	65
Cam focal length f_{cam} ($\times s_{cam}$)	1573
Screen pixel side-length s_t (mm)	0.248
Principal point (o_x, o_y)	(960.1, 539.2)
Noise-variance σ_n^2	101.28
Lens-blur variance σ_{blur}^2 ($\times s_{cam}^2$)	0.25
$\|R_s\|$ ($=\|R_{cam}\|$) (pixels)	1920 × 1080
Focal-distance d_f (m)	0.39

8.2.3. Throughput comparison with existing prototypes

We compare our MIMO capacity estimates ($T_{blk}(\alpha, B)$) with the throughput of existing prototypes of screen–camera communication. In PixNet [3], bits are modulated onto LCD screen pixels that are decoded by an off—the shelf point and shoot camera. PixNet uses OFDM for modulation and adds (255, 243) Reed–Solomon coding for error correction. Consistent with the definition of a block in our model, PixNet uses a block-size of 84 × 84. PixNet was evaluated using a 30 inch LCD screen as the transmitter and 6 MP CCD camera at the receiver, and up-to a maximum distance of 14 m. The authors also reported the throughput from their implementation of QR codes, which we will call QR-P. The QR-P uses a version 5 QR code with a block size of 5 × 5 pixels, and that encodes 864 bits per QR code. On the other hand, COBRA [5] uses color barcodes to communicate between smartphone screen and camera, and was evaluated up-to a maximum distance of 22 cm, and with a block size of 6 × 6 pixels. The authors of [5] have also implemented a smartphone (receiver) version of PixNet, which we will call PixNet-C, where the settings remained the same as original PixNet system.

In Table 1, we report the ratio of throughput from Eq. (7) to the throughput of the these prototypes, for the same parameter settings, of block size and α as in their existing implementations. Our estimates indicate that there is room for at least 2.5× improvement in throughput when compared to capacity. The discrepancy in throughput in these existing prototypes can be attributed to different parameter choices. For example, PixNet uses OFDM modulation and coding which add communication overheads, which have to be incorporated in a limited spatial bandwidth available on the screen. COBRA also incurs loss in throughput due to coding overheads, and additionally the small block size allows for more interference, reducing SINR. COBRA minimizes blur by using repetitive color patterns and intelligent placement of those patterns on the screen. While this strategy minimizes the effect of interference from neighboring pixels, the repetition causes under-utilization of the spatial bandwidth. In general, our findings, supported by these exemplar comparisons, open up interesting questions in the design space for improving information throughputs of screen–camera communication systems.

8.2.4. Motion-blur experiments

To understand the effect of blur alone on the capacity we first plot the measured capacity $C_{campixel}(\alpha)$ at a fixed perspectives (distance of 1 m where $\alpha = 0.5$ and 5 m where $\alpha = 0.5$ and at angle = 0) in Fig. 8. We observe that blur can significantly affect capacity, for example we can observe that the capacity drops drastically when the blur levels are high even when the perspective scaling is only 50%. We observe that the capacity drop is steeper at long distance. We note that a blur kernel of size 1 pixel indicates no blur and at this perspective (α value of 1) the capacity is 6 bits/camera-pixel for the distance and angle between the screen and camera in this experiment.

To understand the effect of motion blur on the signal quality and the effectiveness of de-blurring, we conducted an experiment where we captured a video stream of the screen displaying a chessboard pattern with white and black blocks of size 15 × 15 pixels. During the course of this experiment the camera was hand-held for one case, and the other case, the hand-held camera was intentionally moved (in a horizontal waving pattern) at a nominal speed approximately equivalent to when the user is walking. The distance between the screen and camera was 1 m; at this distance only 50% of camera image is occupied by the screen transmitter pixels. We then applied a Weiner filter based deblurring function available in MATLAB [23] to each of the 100 consecutive images from the video-streams in both cases (see Fig. 9(a)–(d)). Similar to the previous experiments, we then estimated the capacity of screen–camera communication for these two cases by estimating the average perspective distortion factor and the average SINR from the deblurred images. Our estimates indicated a capacity value of 5 bits/camera-pixel for the hand-held case and about 2 bits/camera pixel for the motion case. With reference to Fig. 8 our findings from this experiment indicates that even when the camera is hand-held the capacity of screen–camera communication can be reached as close to as it is when the camera is stationary—with the motion de-blurring features available in off the shelf cameras. For the motion case, without de-blurring, the capacity is almost zero due to the large

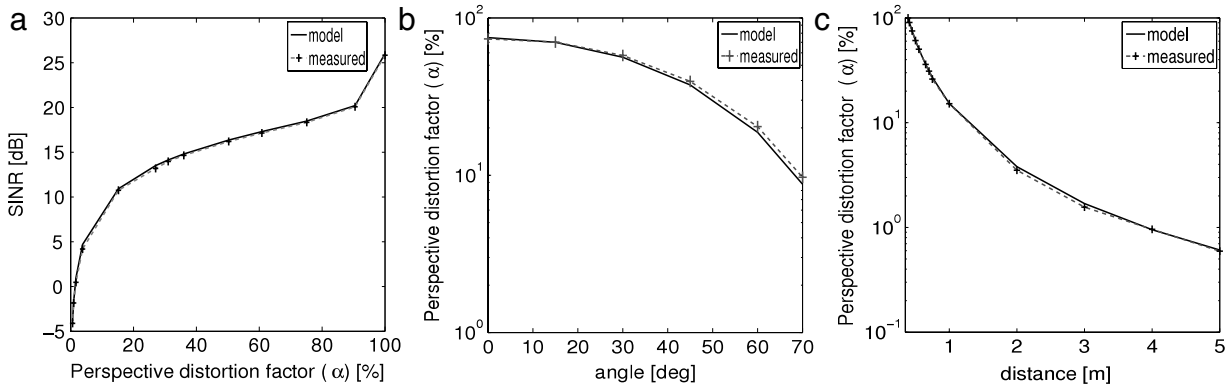


Fig. 7. (a) SINR for different perspective scaling (α) of screen image on camera (b) Perspective distortion α vs. angle between screen and camera (c) Perspective distortion factor α vs. distance between screen and camera.

number of bit errors due high inter pixel interference. However, we observe that de-blurring can help achieve a reasonable data capacity. From this experiment we infer that by using a simple filtering operation the capacity can be improved to a reasonable amount. We also infer, based on Fig. 8, that the amount of blur is approximately 1 pixel for the hand-held case and about 15 pixels for the motion case, in each dimension.

8.3. Perspective distortion factor

The objective of this experiment was to determine the perspective distortion factor α from our measurements to estimate capacity. Since α quantifies the relative area occupancy of the screen in the camera image, we measured the average distortion factor as,

$$\alpha_m = \frac{\|R\|}{\|R_{cam}\|} \frac{1}{(1 + 4\sigma^2_{blur})} \tag{8}$$

where $\|R\|$ represents to the total number of camera pixels that correspond to the imaged screen pixels, and R_{cam} is the resolution of the camera. In Fig. 7(b) and (c) we plot α_m as a function of angle and distance, respectively. As can be seen from these plots the measured spatial-bandwidth fits well with the model (maximum error margin of 1.5%). The α_m reported here is the perspective distortion factor for our LCD—tablet (camera) channel. The distance and angle at which $\alpha_m = 0$ in these plots can be construed as the communication range of a system with the same screen and camera parameters. For example, for a screen with 10× the size (a billboard [24]) the distance range is close to 10× (about 40 m) that of our experimental system.

8.4. Signal-to-interference noise ratio

To facilitate capacity estimation, we measured the signal-to-interference noise ratio $\text{SINR}_{\alpha meas}$ in our experimental system.[6]

We plot $\text{SINR}_{\alpha meas}$ vs. α, along with the analytical SINR_α from Eq. (5), in Fig. 7(a). We can observe from that our SINR measurements are in close agreement with our model (maximum error margin of 1.5 dB). We plot the per-block measured SINR $\text{SINR}_{blk}(\alpha, B)$ using $\text{SINR}_{\alpha meas}$ vs. α for different block-sizes B in Fig. 6(c).

We can infer from Fig. 6(c) that, larger the block higher is the per-block SINR. We can also observe that for a block-size $B = 1$, though it provides large number of parallel channels for multiplexing, the signal energy on each channel is much lower than the noise level, even for medium values of α. In this case, additional signal processing is necessary at the receiver which can help decode the low SINR signal with minimal errors. In general, the size of blocks becomes a primary design choice as it affects SINR performance.

9. Conclusion

In this paper, we discussed the applicability of cameras for communication where cameras could be used as receivers for data transmitted in the form of time-varying 2D barcodes from display screens. We modeled a screen–camera channel using camera projection theory, particularly addressing perspective distortions in more detail than prior works. We modeled

[6] We encourage the reader to refer to our conference paper for the details on the measurement procedure.

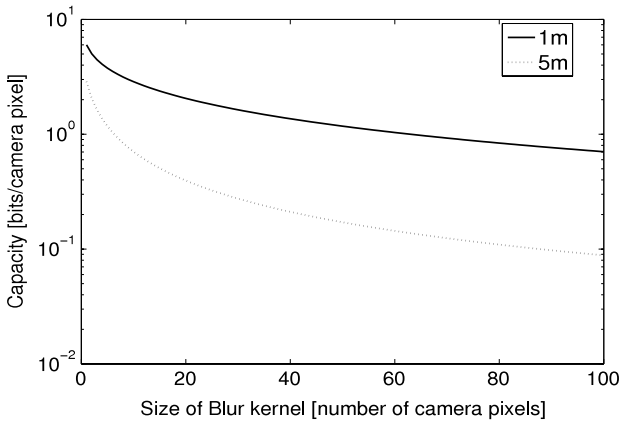
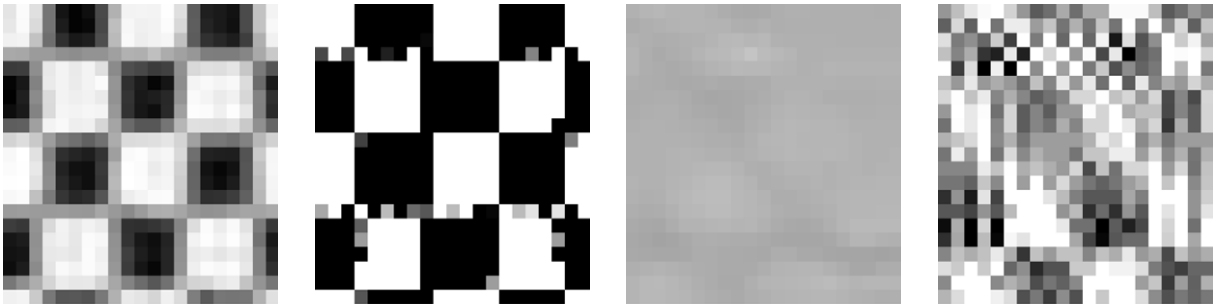


Fig. 8. Capacity vs. blur.



(a) Handheld. (b) Deblurred handheld. (c) Motion. (d) Deblurred motion.

Fig. 9. Illustration of motion blur and de-blurring on images taken by a hand-held camera ((a), (b)), and when camera is in motion ((c) and (d), phone was hand-held and swayed from left–right at nominal speed).

and studied the effect of perspective distortion on the information capacity of screen–camera communications. Through extensive in-lab calibration experiments we found that, even with the frame-rate limitations in off-the-shelf mobile cameras, data-rates of the order of hundreds of kbps-to-Mbps is possible even when the 2D barcode from the screen images onto only a small portion of the camera image. Our experimental capacity bound was also in good agreement with the theoretical bound. Our findings indicate that camera communications is still promising for medium sized data-transfer or even streaming applications; such as downloading a file from a smartphone screen or streaming a movie from a large display wall. Our estimates indicate that current prototypes have only achieved less than half their capacity, which means that designing efficient techniques to address perspective distortions is still an open problem for building high-data rate camera communications.

Acknowledgments

We thank all the anonymous reviewers and the editor of PMC journal for their valuable suggestions and comments. This work was supported by the US-National Science Foundation (NSF) under the grant CNS-1065463.

References

[1] A. Ashok, M. Gruteser, N. Mandayam, J. Silva, M. Varga, K. Dana, Challenge: Mobile optical networks through visual MIMO, in: Proceedings of MobiCom, ACM, New York, NY, USA, 2010, pp. 105–112.
[2] IEEE P802.15 Working Group for Wireless Personal Area Networks: On Study Group Status for Camera Communications. http://tinyurl.com/odkrr9w.
[3] S.D. Perli, N. Ahmed, D. Katabi, PixNet: Interference-free wireless links using LCD-camera pairs, in: Proceedings of MobiCom'10, ACM, New York, NY, USA, 2010, pp. 137–148.
[4] S. Hranilovic, F. Kschischang, A pixelated-MIMO wireless optical communication system, IEEE J. Sel. Top. Quantum Electron. 12 (4) (2006) 859–874.
[5] T. Hao, R. Zhou, G. Xing, COBRA: color barcode streaming for smartphone systems, in: Proceedings of MobiSys'12, ACM, New York, NY, USA, 2012, pp. 85–98.
[6] H. Chen, R. Sukhthankar, G. Wallace, T. Jen Cham, Calibrating scalable multi-projector displays using camera homography trees, in: CVPR, 2001, pp. 9–14.
[7] R. Yang, D. Gotz, J. Hensley, H. Towles, M.S. Brown, PixelFlex: A reconfigurable multi-projector display system, 2001.
[8] X. Liu, D. Doermann, H. Li, A camera-based mobile data channel: capacity and analysis, in: Proceedings of MM, ACM, NY, USA, 2008, pp. 359–368.
[9] HCCB High Capacity Color Barcodes. http://research.microsoft.com/en-us/projects/hccb/about.aspx.
[10] A. Goldsmith, Wireless Communications. Cambridge, 2005.

[11] T.S. Lomheim, G.C. Holst, CMOS/CCD Sensors and Camera Systems, second ed., The International Society for Optical Engineering (SPIE), 2011.
[12] A. Tang, J. Kahn, K.-P. Ho, Wireless infrared communication links using multi-beam transmitters and imaging receivers, in: ICC 96, Conference Record, Converging Technologies for Tomorrow's Applications, vol. 1, June 1996, pp. 180–186.
[13] T. Komine, M. Nakagawa, Fundamental analysis for visible-light communication system using LED lights, IEEE Trans. Consum. Electron. 50 (1) (2004) 100–107.
[14] B.K.P. Horn, Robot Vision, MIT Press, Cambridge, MA, USA, 1986.
[15] A. Ashok, S. Jain, M. Gruteser, N. Mandayam, W. Yuan, K. Dana, Capacity of pervasive camera based communication under perspective distortion, in: Proceedings of PerCom, IEEE, 2014.
[16] R.I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, second ed., Cambridge University Press, ISBN: 0521540518, 2004.
[17] Lecture notes by D.V. Valen: Point Spread Function Workshop. http://tinyurl.com/p88lkbr.
[18] S.K. Nayar, M. Ben-Ezra, Motion-based motion deblurring, IEEE Trans. Pattern Anal. Mach. Intell. 26 (6) (2004) 689–698.
[19] Jia Chen, Lu Yuan, Chi-Keung Tang, Long Quan, Robust dual motion deblurring, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008, IEEE Conference on, June 2008, pp. 1–8.
[20] L. Zheng, D.N.C. Tse, Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels, IEEE Trans. Inform. Theory 49 (2003) 1073–1096.
[21] Camera calibration toolbox for MATLAB. http://www.vision.caltech.edu/bouguetj/.
[22] Apple—IPhone 5s—iSight Camera. http://www.apple.com/iphone-5s/camera/.
[23] Weiner Deblurring. http://www.mathworks.com/help/images/ref/deconvwnr.html.
[24] Billboard sizes. http://www.sbuilts.com/sizes.cfm.