Master Thesis

Imperceptible AR Markers for Near-screen Pointing Interaction Using Smartphones

（スマートフォンを用いて近距離からディスプレイと
ポインティング連携するための不可視 **AR** マーカ）

January 30th, 2020

Department of Information and Communication Engineering
Graduate School of Information Science and Technology

48-186402

Akira Matsumoto

Supervisor: Professor Takeshi Naemura

Abstract

Owing to the pervasive use of displays and smartphones, mobile interactions with display screens have gained attention within the advertising and gaming industries as well as in human–computer interaction research. Communication through QR code-like markers and localization via AR markers are common examples of such interactions. However, these visible markers interfere with the display content; this problem is critical for localization over a wide range of interactions, and fewer markers result in less reliability and accuracy. Although some studies have addressed this issue, few have focused on near-screen interaction without additional hardware. To address this problem, we propose an easy-to-install localization method that uses an array of AR markers, which are made imperceptible to the human eye through chromaticity vibration at 30 Hz. We mainly focus on applications, such as digital signage, where users point their smartphones at the display content. Through four evaluations, we confirm that the pointing error is within 1 mm, and that the proposed system works, when the distance between the screen and smartphone is 4–24 times the size of the AR marker. In addition, we establish that our system is robust against rotation. Finally, we present two potential application scenarios, advertising and navigation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Owing to the widespread use of smartphones and the increase in the number of public and private displays, the interaction between display screens and smartphones has attracted attention in the fields of human–computer interaction (HCI) and advertising. In this mobile interaction with screens, the seamless connection of devices is crucial. The currently prevalent systems mainly involve a scenario in which smartphones are only used as gateways for the input and output operations of users, and hardly use the positional relationship between these devices.

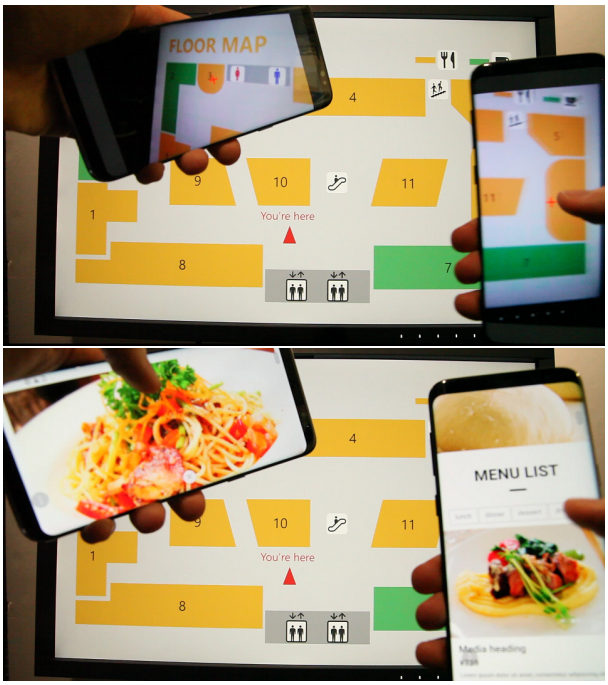
Some researchers have focused on device interaction as a space-aware interaction (interaction that varies according to the positional relationship between the screen and smartphone). In this interaction, it is essential to localize the smartphone and measure the positional relationship between the devices. This measurement enables intuitive operation (e.g., a smartphone as a pointer or mouse on the display), enriching the interaction [4, 24, 39].

For measuring the relationship between the smartphone and screen, the capturing of AR markers using the smartphone camera is a convenient method [9, 29, 32]. However, the display of visible markers impairs the users' visual experience because they occlude the display contents [25, 29]. To solve this problem, various methods have been proposed. By using special hardware, a wide range of interactions can be realized; however, the installation is inconvenient [8, 19, 36]. Feature tracking can be realized by off-the-shelf hardware [5, 7], but the display content is limited to those containing rich features. Yamamoto *et al.* [48] embedded random dot markers [41] on screens utilizing imperceptible color vibration, which displays two different colors alternately. Color vibration can be generated in ordinary 60-Hz displays. However, approximately 20% of the random dot markers must be captured for reliable tracking [50], preventing closer interaction.

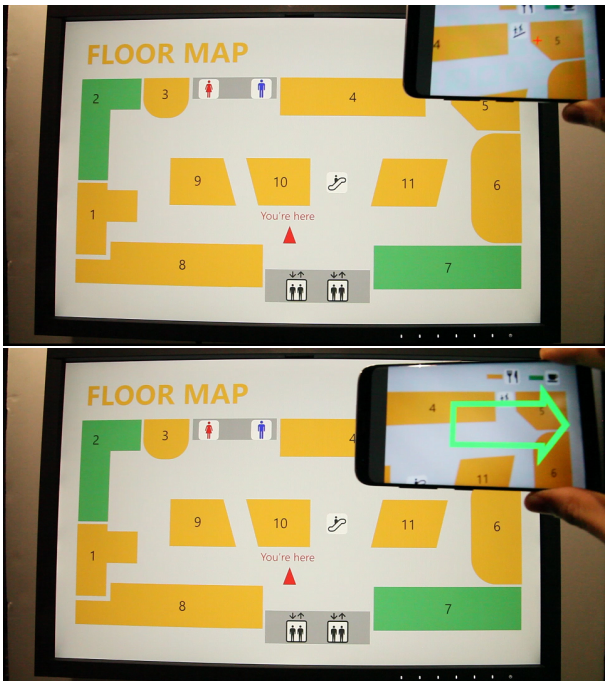
In view of the above, we utilize imperceptible color vibration to embed an array of AR markers for smartphone localization at a closer distance in space-aware mobile interaction with screens, in this study. We define this type of interaction as near-screen interaction. We mainly focus on applications, where a smartphone is pointed at the display content, in near-screen interaction such as digital signage applications. The main motivation for realizing near-screen interaction is because in some cases, users view signage (e.g. maps) from close proximity, and it is convenient for them to receive information in their smartphones because they can continue to access this information after moving away from the screen. In this case, we realize interaction using the positional relationship between the screen and smartphone by measuring the position pointed to by the smartphone in the display coordinate system. We measure the display-pointing accuracy, and clarify the relationship between the marker size and the distance over which marker detection works. We determine, whether our system works, even if the smartphone is tilted or moving. Moreover, we develop a sample application for demonstrating the proposed method, as shown in Figure 1.1.

In summary, the main contributions of this study are as follows:

- Mapping of the related systems in space-aware mobile interaction with screens, considering the ratio of the images to be captured and the advantages, and the presentation of the challenges to be addressed.
- Realization of smartphone localization by embedding an array of AR markers in the display content using imperceptible color vibration, and development of a method for designing the AR-marker size based on the imaging range and angle-of-view of the smartphone camera.
- Evaluation of the proposed system under laboratory conditions, for clarifying the pointing accuracy and working environment. The results show that our marker-size design method is reasonable, and our system is robust against camera shake and tilt.
- Demonstration of the user experience with a prototype application, using a public display and a smartphone.



(a) Advertising



(b) Navigation

Figure1.1: Application scenarios: By pointing the smartphone at a restaurant on the map (first frame), the user can (a) download the menu or (b) be guided to the restaurant (second frame).

Chapter 2

Related Work

There have been a lot of research that propose techniques to use display devices not only to display the mere image but also to output arbitrary optical information with division and multiplexing of time and space. There are several terms for defining the field such as Display-based Computing (DBC) [40] and screen-camera communication. But the scopes of these terms are somewhat ambiguous. So in this paper, we simply divide the field by receiver: screen-sensor method and screen-camera method. Both sections are further divided according to whether they use markers perceptible to the human. Next, we describe research that is especially relevant to our work, that is, research exploring the interaction between screens and smartphones utilizing positional relationship. We divide these research into two by the distance between screens and smartphones which the system works. Finally, we describe the method of embedding and extracting matrix barcode utilizing imperceptible color vibration which we use in our method.

2.1 Screen-Sensor Method

2.1.1 Visible Method

Augmented Coliseum [21, 40] projects visible markers (Figure [21]) on photo sensors put on top of mobile robots and measures the position and direction of the robots (Figure 2.2). However, this system indicates only the relative coordinates so initialization is unavoidable.

Lumitrack [46] projects structured light patterns called m-sequences on linear optical sensors (Figure 2.3). An m-sequence is a binary sequence whose every consecutive subsequence of m bits is unique [27]. Six degree of freedom tracking is available by combining multiple sensors. The main problem of these visible methods is that light transmitting information is obtrusive to human.

2.1.2 Invisible Method

It is known that the maximum flicker frequency perceptible to the human eye is approximately 60 Hz and it is called Critical Flicker Frequency (CFF) [33]. When the light is blinking at frequencies higher than CFF, the flicker is imperceptible to the human eye and only time-averaged luminance is perceived. Kimura *et al.* proposed a method embedding independent data into each pixel of an image as high-speed flicker by a DLP projector [20]. This method is called pixel-level visible light communication (PVLC). Figure 2.4 illustrates the principle of PVLC. Many applications utilizing PVLC have been proposed. EmiTable [20] is a smart tabletop surface with small LED displays which display images according to their position (Figure 2.5). Phygital Field [11] is a system controls a swarm of robots on a projected image (Figure 2.6).

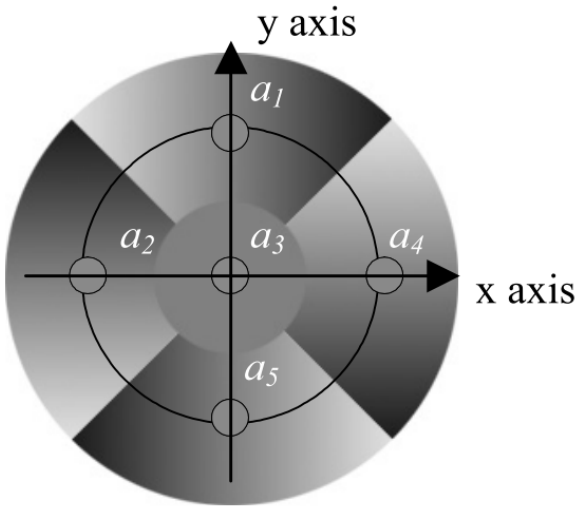


Figure2.1: A fiducial marker used in Augmented Coliseum [21]. a_i is the position of each sensor.

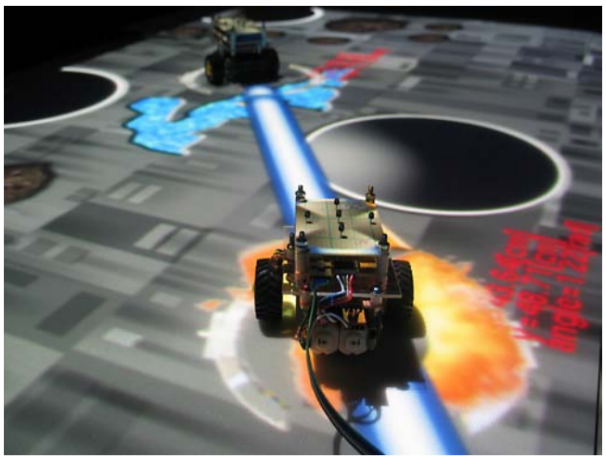


Figure2.2: Augmented Coliseum [40].

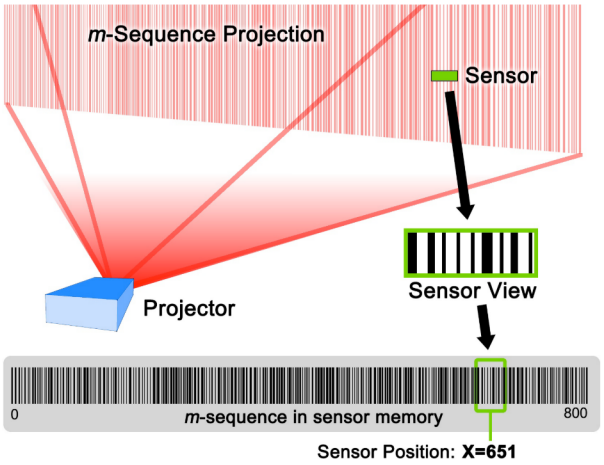


Figure2.3: System overview of Lumitrack [46].

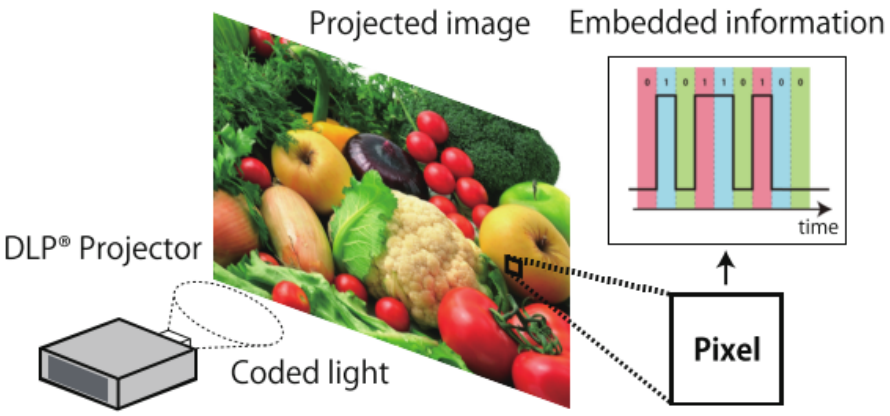


Figure2.4: The principle of PVLC.

Figure2.5: EmiTable [20].

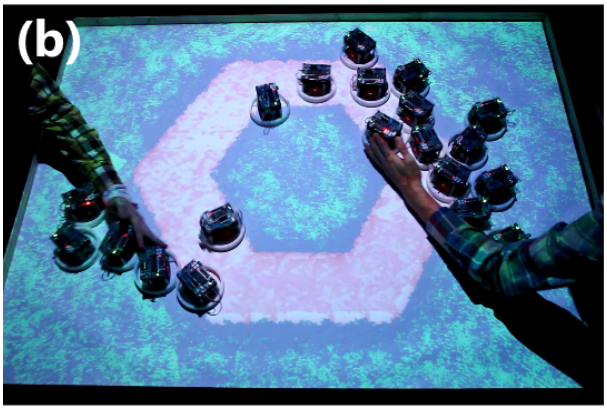


Figure2.6: Phygital Field [11].

Figure2.7: Prakash [31].

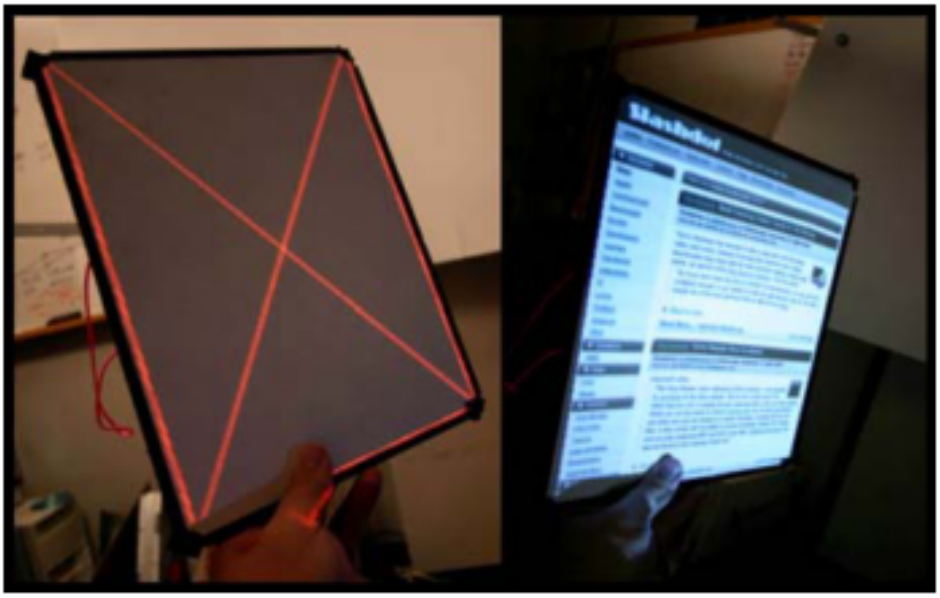


Figure2.8: Tracking the location of a hand-held surface and then projecting content [22].

Prakash [31] is a high speed optical motion capture method realized by projecting gray coded patterns (Figure 2.7). It transmits position data by aligned infrared LEDs with passive films. Each LED represents one bit position of the binary Gray code and flashes in order. Photosensors acquire position and orientation from the lights.

Lee *et al.* converted a light source of a DLP projector to red and infrared LEDs [22]. They projected gray-coded binary structured light patterns [23] as location data by infrared LEDs and visible image by red LEDs. Figure 2.8 shows the application where a hand-held surface is being tracked and projected onto. There are infrared sensors on the surface and they detect the structured light patterns and report their locations back to the projector.

Abe *et al.* [2] embedded imperceptible information into ordinary 60 Hz display utilizing

Figure2.9: The smartphone plays the video in accordance with the thumbnail [2].

imperceptible color vibration and decoded using a photosensor. They embed five different data values using the vibration of each RGB channel. Figure 2.9 shows the application that different data values are embedded in four thumbnails respectively and the smartphone with a photosensor plays the video in accordance with the thumbnail.

2.2 Screen-Camera Method

2.2.1 Visible Method

The most common example of data transmission using screens is QR code [15]. It displays a two-dimensional (2D) black and white pattern. Although there are some similar 2D barcodes [13, 14, 16], the QR code is most widely used.

In the research area, a lot of barcodes which boost data capacity and robustness have been proposed. PixNet [30] proposes orthogonal frequency division multiplexing (OFDM)-based matrix barcode to boost data capacity and robustness. OFDM is the transmission scheme widely used in radio frequency (RF) technologies. Unlike RF-based OFDM schemes that encode data in time frequencies, PixNet encodes data in 2D spatial frequencies of luminance.

COBRA [10] and RainBar [44] densifies the data by aligning colored cells (Figure 2.10). They utilize color as another data channel compared to the QR code. COBRA realizes high throughput with lower computational resource than PixNet.

LightSync [12] deals with synchronization between the screen and camera. It proposes a

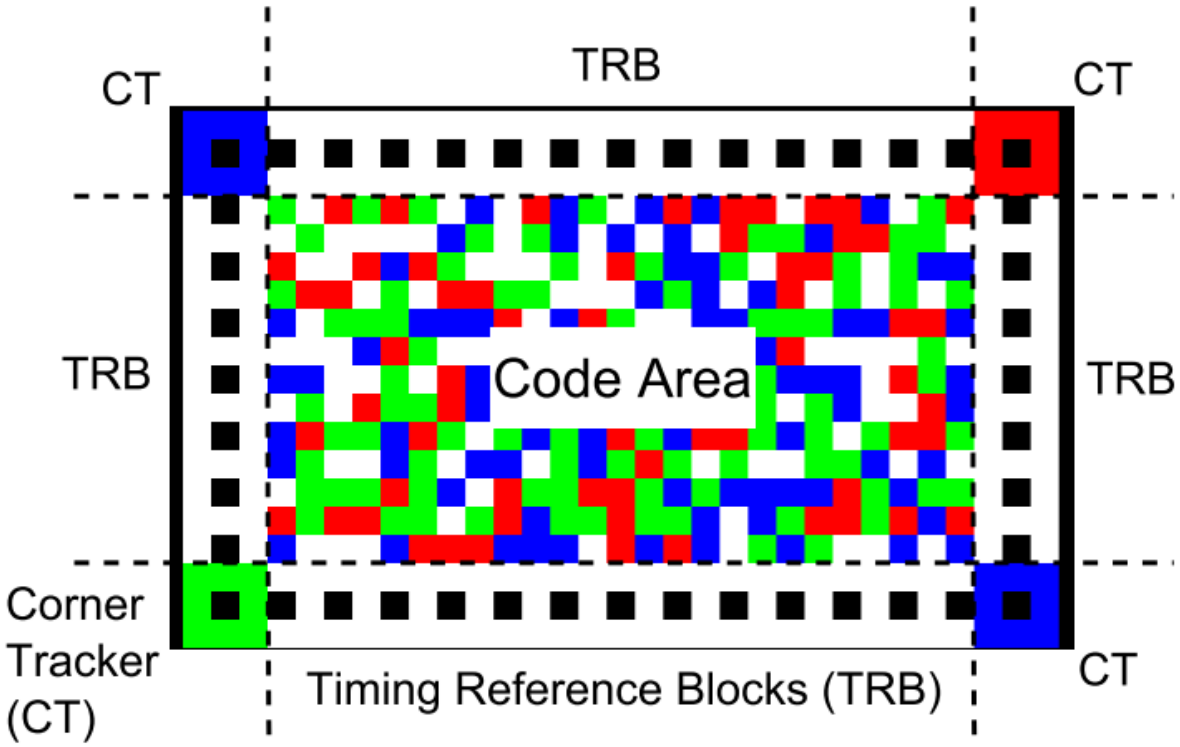


Figure2.10: Design of COBRA [10].

synchronization method applicable to existing 2D barcodes, independent from their design. A typical method to capture each frame of the screen is to set the camera's frame rate to the double of display's refresh rate, but LightSync requires the camera's frame rate to be only half the display's refresh rate. LightSync adds several fields to the existing barcode to achieve synchronization (Figure 2.11).

ShiftCode [53] encodes data bits with shifting shape patterns (Figure 2.12). It dealt with all the issues of throughput, synchronization, and reliability at the same time. It has the highest throughput among the codes we mentioned above, does not need synchronization between transmitter and receiver, and has high reliability.

2.2.2 Invisible Method

InFrame [43] uses the vibration of luminance to embed data to video contents. It splits a display into blocks, assigning 1 or 0 by vibrating or not (Figure 2.13). As mentioned in section 2.1.2, CFF is about 60 Hz so this system requires a 120-Hz display, which is hard to say prevalent. The captured frame is smoothed and the difference of it from the original frame is used to acquire bits.

InFrame++ [42] improved the idea of InFrame. While InFrame simply displayed luminance-

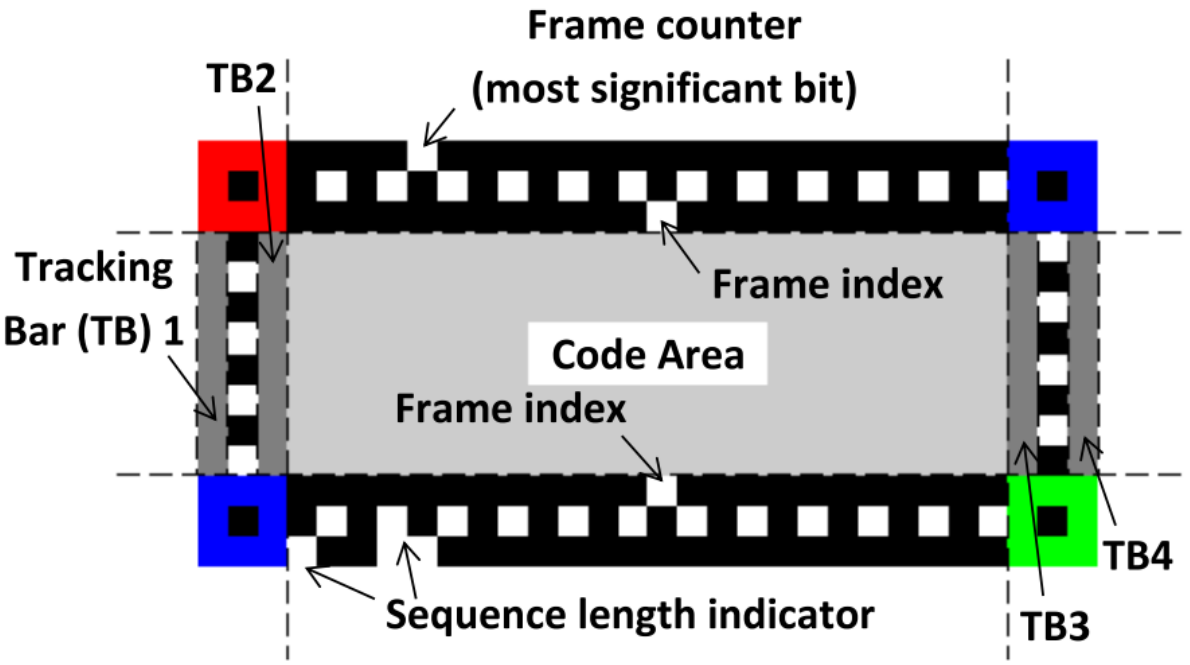
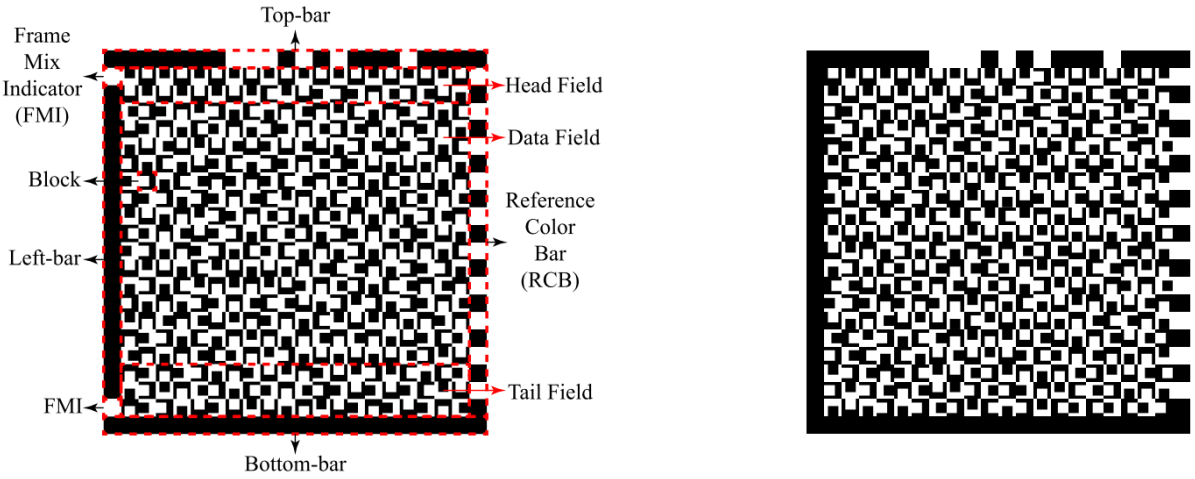


Figure2.11: LightSync codes added to COBRA [12].



(a) Even frame. (b) Odd frame.

Figure2.12: Design of ShiftCode [53].

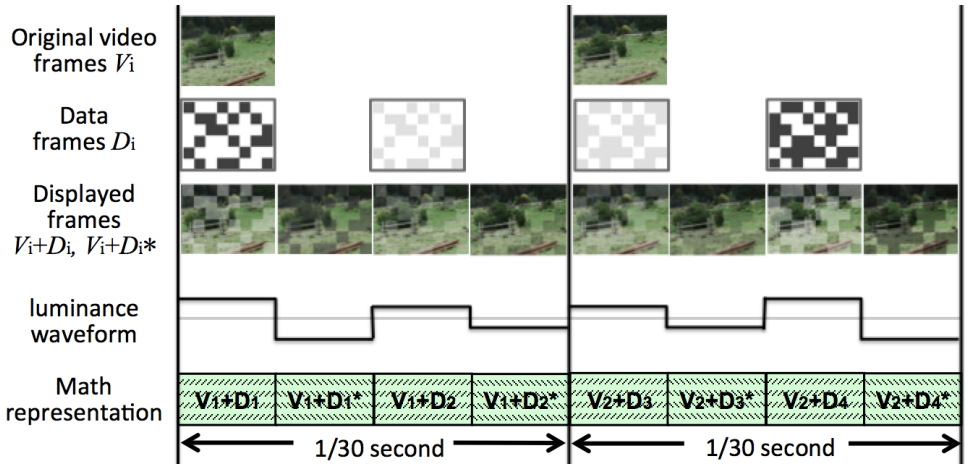


Figure2.13: Data embedding of InFrame [43].

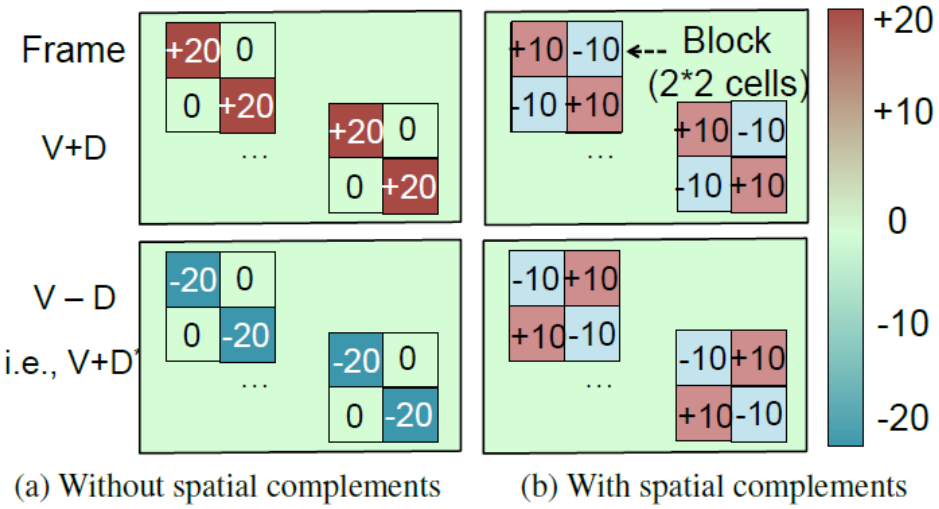


Figure2.14: Data embedding of (a) InFrame and (b) InFrame++ [42].

added frame and luminance-subtracted frame alternately, InFrame++ aligns both added and subtracted cells in the same frame (Figure 2.14). This is effective for suppressing the visibility of data embedded. Furthermore, it introduced a Code Division Multiple Access-like modulation scheme and the locator of the QR code for robust decoding.

HiLight [26] embeds data into images by modulating pixel translucency. Modulating α values require less computational time than RGB values like InFrame. It divides the screen into grids and represents 1 and 0 by translucency change at 20 Hz and 30 Hz respectively (Figure 2.15). The receiver needs to be able to capture at least 60 fps and applies Fast Fourier Transform for decoding.

There are some other researches that tackle with imperceptible data transmission, but most of them use original marker pattern and full frame of the image [18,28,38,52]. Abe *et al.* [1,3]

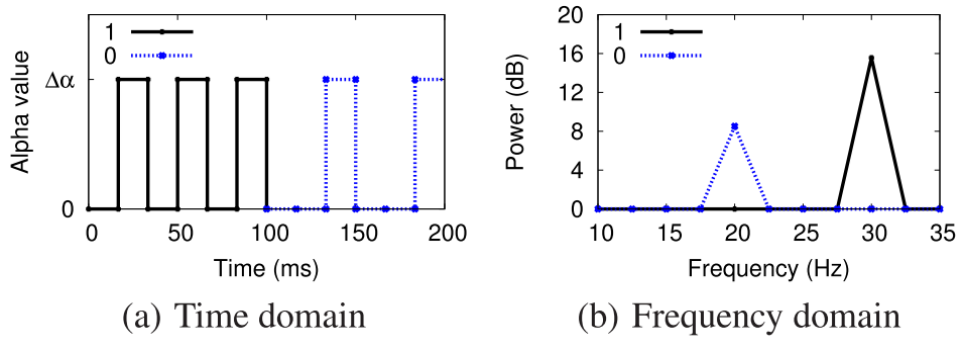


Figure2.15: HiLight embeds bits into the frequency of α value vibration [26].

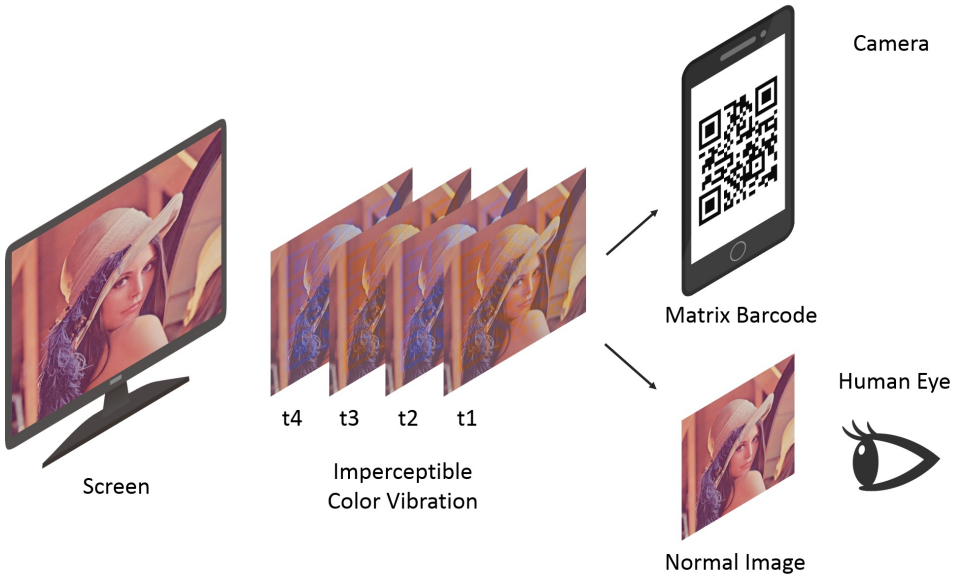


Figure2.16: System operation of [1].

proposed a methodology of embedding an arbitrary matrix barcode into ordinary 60 Hz display utilizing imperceptible color vibration and decoding with a smartphone camera (Figure 2.16). Detail of the system will be explained in section 2.4.

2.3 Space-aware Mobile Interaction with Screens

Some studies have explored the space-aware interaction between screens and smartphones. There are various methods for tracking smartphones, which determine the distance between the smartphone and screen. These methods can be roughly divided into two types: near-screen and far-range. Figure 2.17 compares the related work on space-aware mobile interaction with screens in terms of 1) the ratio of the image to be captured, and 2) three advantages: the unobtrusive feature (UF), off-the-shelf hardware (OH), and content independency (CI).

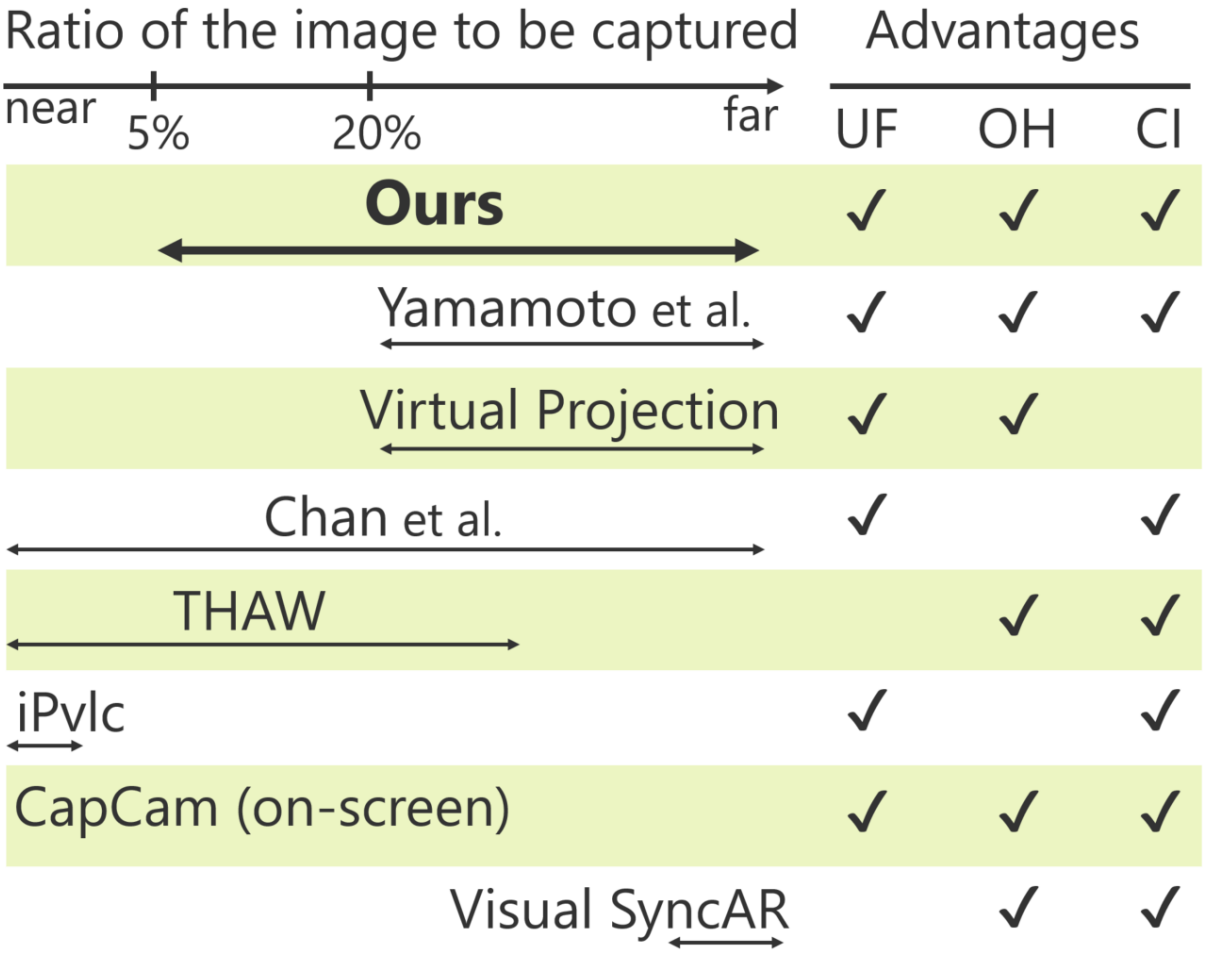


Figure2.17: Comparison of the related work on space-aware mobile interaction with screens. The advantages of the respective systems are mapped as per three categories: unobtrusive feature (UF), off-the-shelf hardware (OH), and content independency (CI).

Figure2.18: iPvlc [19].

Figure2.19: THAW [25].

2.3.1 Near-screen Interaction

The iPvlc realizes interaction between a screen and a smartphone placed on the screen (Figure 2.18) [19]. This system transmits position information utilizing pixel-level visible light communication (PVLC) [20], and the smartphone receives information through photodetectors. Although this method realizes precise tracking, it requires special hardware.

The THAW [25] tracks a smartphone that is placed on or hovered over a screen by displaying a 2D color pattern in the camera's field-of-view alone (Figure 2.19). This system requires only off-the-shelf hardware. Although its color pattern is meant to be occluded by the smartphone, it is impossible to hide the pattern completely.

The CapCam [47] uses a touchscreen for tracking (Figure 2.20). In addition, it automatically

establishes a wireless link between the smartphone and screen by displaying a sequence of color to the smartphone rear camera. Although this system is completely independent of the displayed content, the smartphone needs to touch the screen. There are several studies using a touchscreen but most of them have similar limitations [37, 51].

Figure2.20: CapCam [47]

2.3.2 Far-range Interaction

Chan *et al.* [8] combined an infrared and ordinary color projector to project visible content and invisible markers, simultaneously (Figure 2.21). This system projects AR markers through an infrared projector and detects the markers using an infrared camera connected to a smartphone. It dynamically changes the marker size such that the camera can detect the markers from various distances. Although this approach realizes a wide range of interaction by changing the marker size dynamically, special hardware, such as the infrared projector and infrared camera, are undesirable because they are difficult to install.

Virtual Projection [5] tracks a smartphone by detecting the feature points [6] in the displayed image (Figure 2.22). While this system does not require obtrusive markers or special hardware, its display content is limited to that containing rich features, i.e., tracking depends upon the display content. Moreover, the smartphone needs to capture 20% of the image for reliable tracking.

Visual SyncAR [49] surrounds the display content with a white frame, and tracks a smartphone by detecting the frame (Figure 2.23). This system sends a timestamp of the content through a digital watermark. It vibrates the pixel values of the content slightly such that the

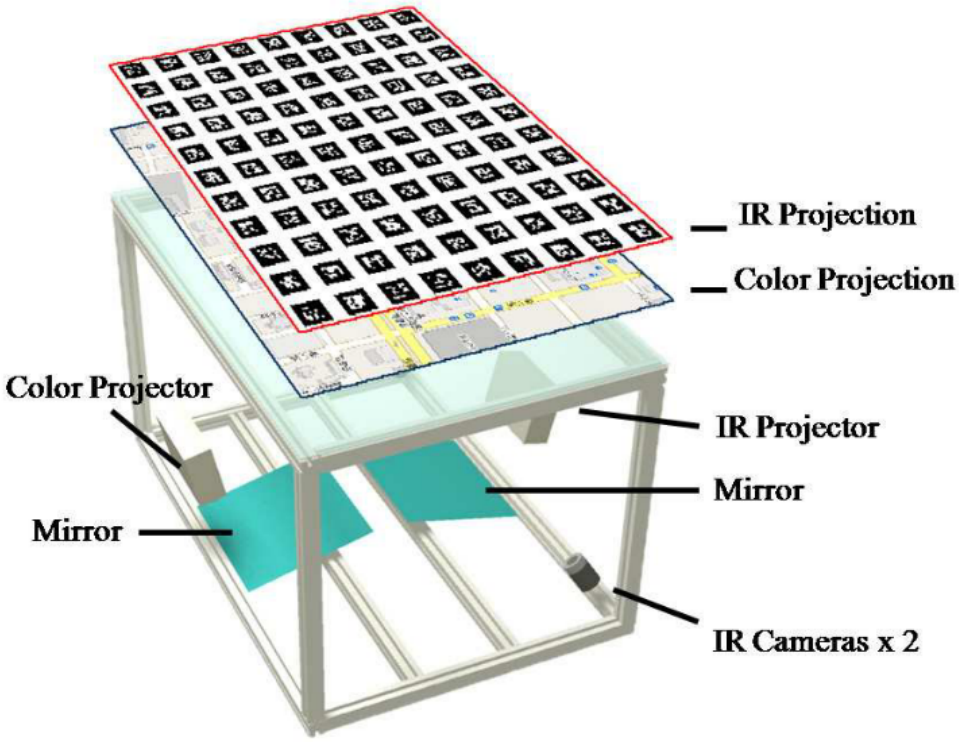


Figure2.21: The hardware architecture of the system [8].

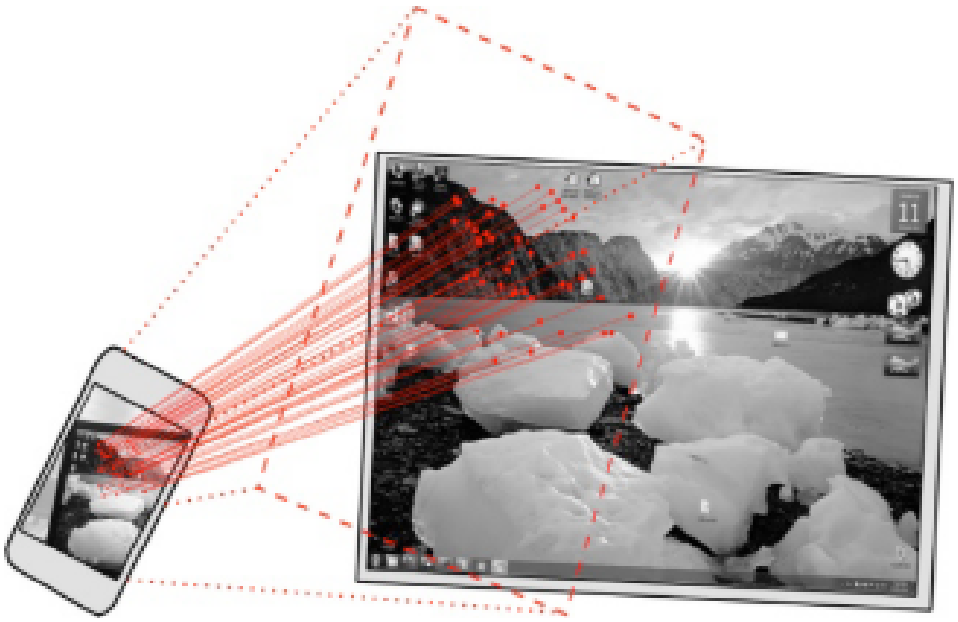


Figure2.22: Content-based image tracking process [5].

Figure2.23: Visual SyncAR [49].

human eye cannot distinguish the modulated images. Although this system does not depend upon the display content, the smartphone has to capture the entire display for tracking.

Yamamoto *et al.* [48] embedded random dot markers [41] on screens utilizing imperceptible color vibration (Figure 2.24) (color vibration will be explained in detail in the next section). Although this system uses an ordinary 60-Hz display and does not need to capture the entire display, approximately 20% of the random dot markers must be captured for tracking [50].

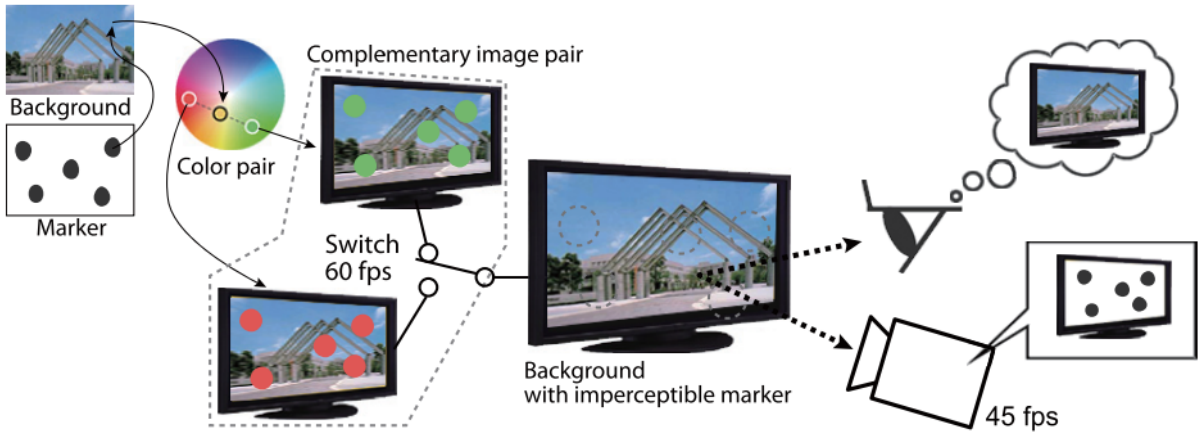


Figure2.24: An overview of the method [48].

2.4 Imperceptible Color Vibration

Imperceptible color vibration is a promising method for embedding invisible code into images, in the temporal domain. The maximum chromatic flicker frequency perceptible to the human eye is approximately 25 Hz [17], which can be generated by an ordinary 60-Hz display. Several studies have utilized imperceptible color vibration [3, 45, 48]. The system proposed by Yamamoto *et al.* [48] is unsuitable for our purpose because it uses a webcam and desktop PC for decoding. When digital signage is used, it is preferable, if the receiver is a device usually carried around by people. Abe *et al.* proposed a method for embedding matrix barcodes into images utilizing imperceptible color vibration, and extraction using a smartphone camera [1,3]. This system satisfies our requirements because the receiver is a smartphone.

To transmit information without impairing the display content, Abe *et al.* modulated the original color of each pixel with two colors that have the same luminance as the original [1]. Imperceptible color vibration can be generated by displaying these two colors alternately, using ordinary 60-Hz displays. The code pattern can be embedded either by vibrating or not vibrating each pixel, thereby representing black or white, respectively. For decoding, a simple solution is to capture a video at 120 fps using a smartphone. Then, the two modulated colors alternately appear in the captured frames, as shown in Figure 2.25. However, this is not practical because of the restriction of the Android Platform. Currently, capturing and processing images in real time when captured at faster than 30 fps is unavailable. To solve this problem, they record a video at 24 fps and set exposure time to 1/120 s (Figure 2.26). Then the idle time becomes 1/30 s (vibration cycle) so the two modulated colors alternately appear just like when captured at 120 fps. The vibrating pattern can be extracted by considering the difference between the two frames, and thresholding the output.

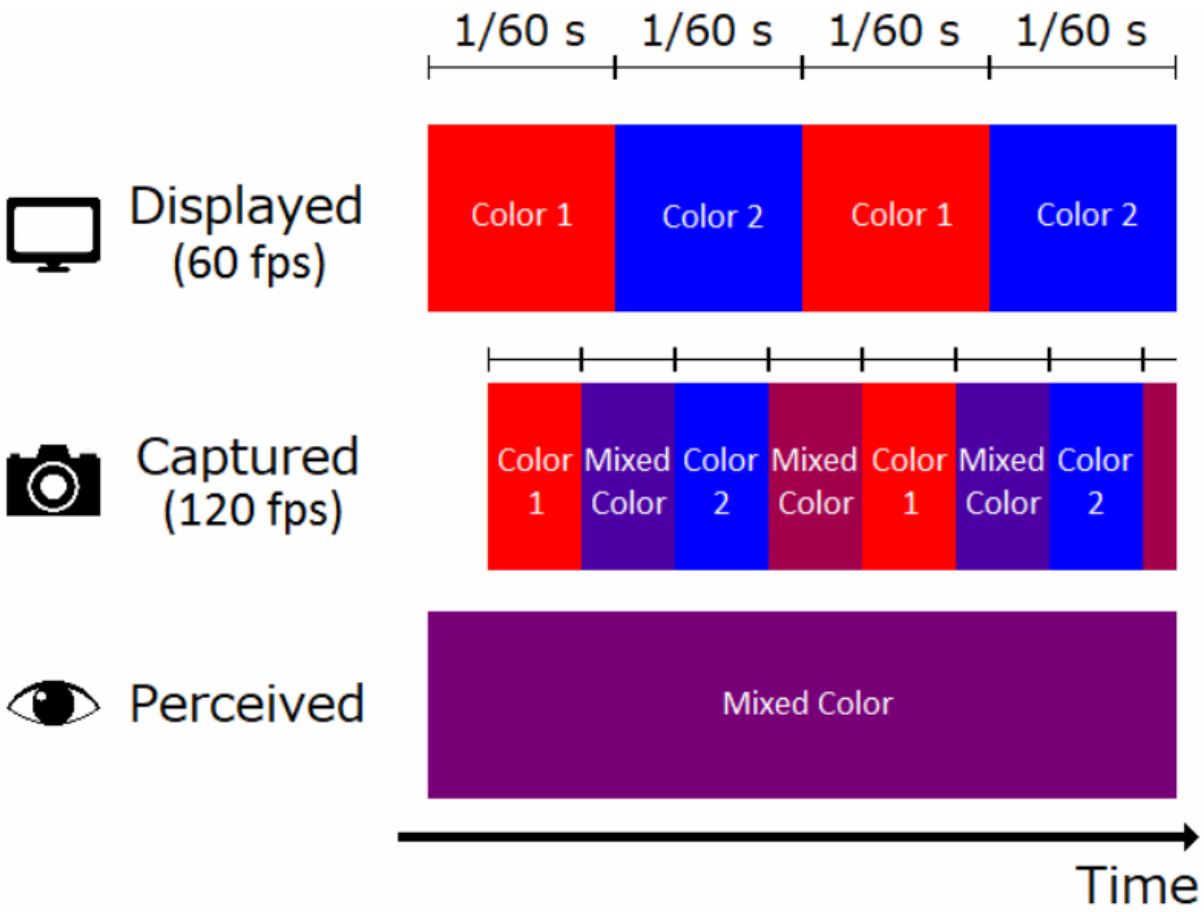


Figure2.25: Time sequences of the frames for the system components and human eye when captured at 120 fps [3].

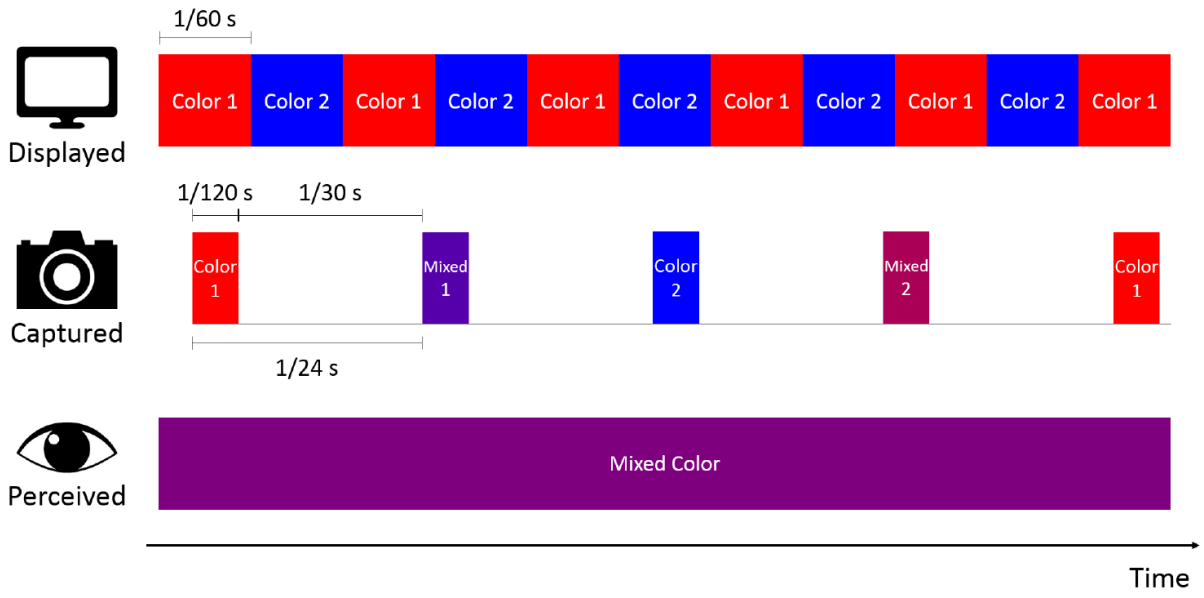


Figure2.26: Time sequences of the frames for the system components and human eye when captured at 24 fps [1].

Chapter 3

Method

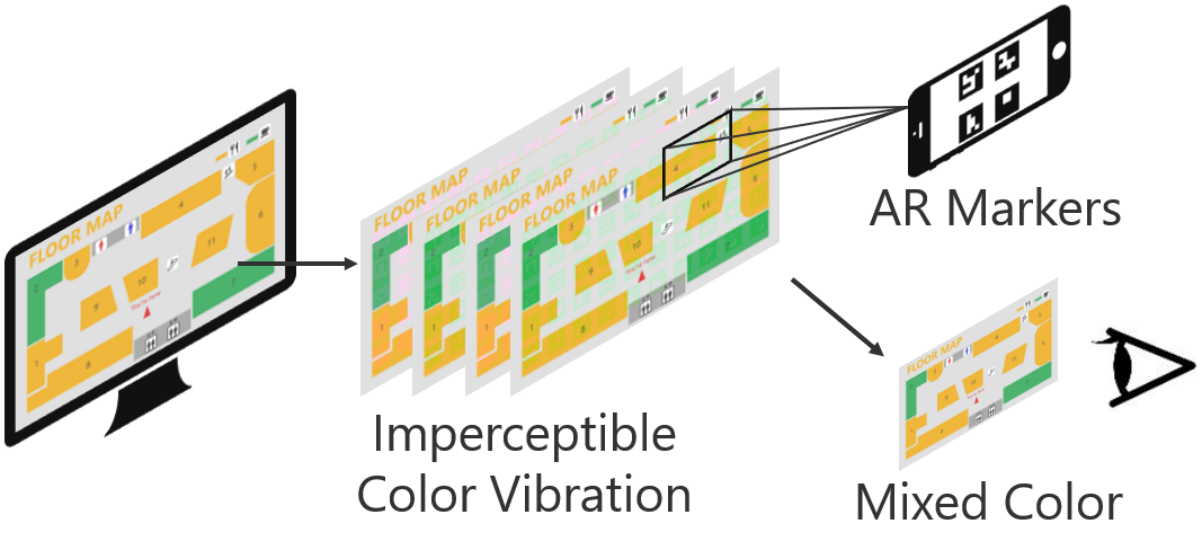


Figure3.1: System operation with the proposed method. Two color-vibrated images are displayed alternately. The smartphone camera extracts the embedded AR markers, while the human eye perceives the normal image alone.

We propose a smartphone localization system by embedding an array of AR markers into images using imperceptible color vibration, and a method for designing the marker size based on the imaging range and angle-of-view of the smartphone camera. As shown in Figure 3.1, we embedded an array of AR markers into the display content using imperceptible color vibration, and detected them using a smartphone. We can calculate the position of the smartphone from the detected markers. The size of the AR markers and their arrangement determines the maximum and minimum distances of the smartphone from the screen. Let M be the length of the marker-side and G be the interval between markers. We assume that the center of the smartphone camera points to a location within the dashed rectangle, as shown in Figure 3.2. The distance between the rectangular border and the outermost markers is $G/2$. Let θ be the diagonal angle-of-view of the smartphone camera, $m : n$ $(m \geq n)$ be the aspect ratio of the video, and L be the distance between the smartphone and screen. Assuming that the smartphone is parallel to the screen, the area of the screen captured by the smartphone will be a rectangle with a diagonal length of $2L\tan\frac{\theta}{2}$ (the orange rectangle in Figure 3.2). The length of the shorter-side of the rectangle is αL, where $\alpha = \frac{2n\tan\frac{\theta}{2}}{\sqrt{m^2+n^2}}$.

To determine the minimum L needed to capture an entire marker, regardless of how the smartphone is translated and rotated, it is sufficient, if a marker is always present within the circle, centered on the center of the rectangle and inscribed within the rectangle (green circle in Figure 3.2). This condition is satisfied, if the diameter of the circle is greater than the diagonal length of the square surrounding 2×2 markers. Therefore,

$$\alpha L \geq \sqrt{2}(2M + G) \tag{3.1}$$

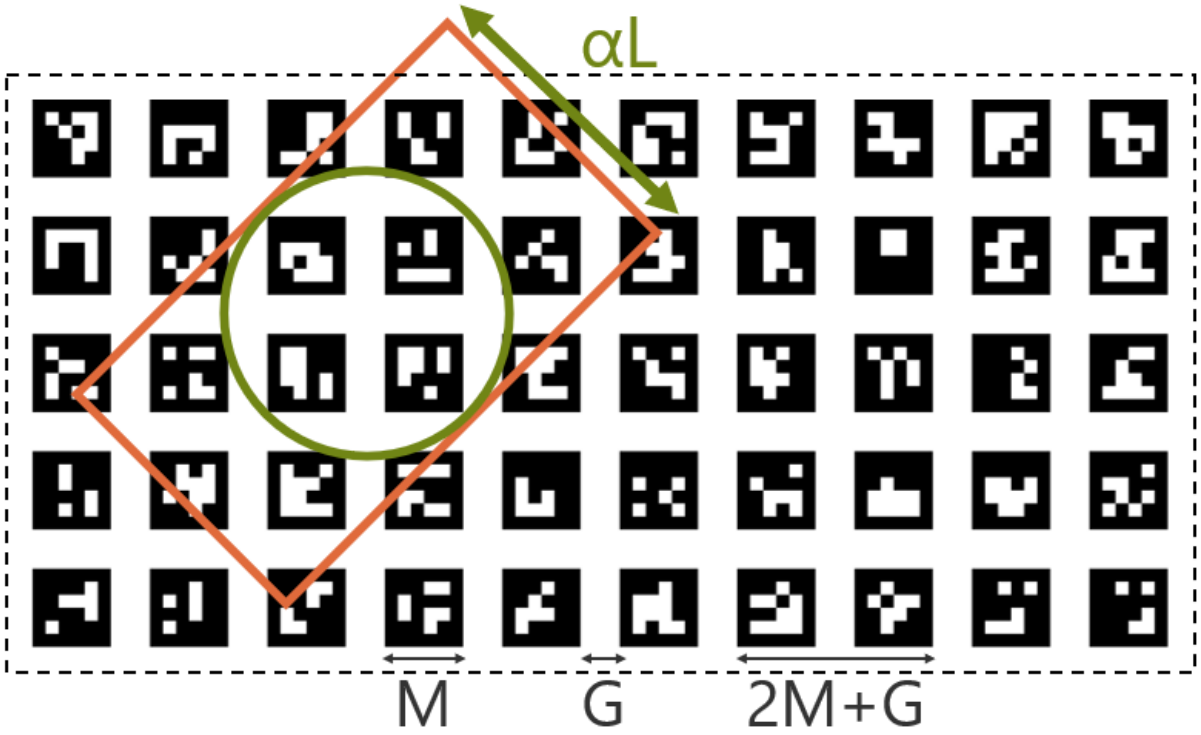


Figure3.2: Marker-alignment example. The orange rectangle indicates the region captured by the smartphone. The green circle indicates the region always captured by the smartphone, regardless of its rotation.

must be satisfied to capture an entire marker.

Next, we consider the maximum L at which camera localization is possible. Let R be the number of pixels of the captured image corresponding to αL, P be the number of pixels of the captured marker corresponding to M, and P_{min} be the minimum P required for detecting the marker. The value of P can be calculated from the product of R and the ratio of M to the length of the shorter-side of the rectangle. Hence,

$$P = R\frac{M}{\alpha L} \geq P_{min}. \tag{3.2}$$

In summary, the range of L is

$$M\frac{R}{\alpha P_{min}} \geq L \geq \frac{\sqrt{2}(2M+G)}{\alpha}. \tag{3.3}$$

The value of P_{min} is calculated experimentally and discussed in the next section. Table 3.1 shows definitions of the variables and Figure 3.3 illustrates the meaning of variables.

Table3.1: Variables definitions.

Variables	Definitions
M	Marker size
G	Marker spacing
θ	Diagonal angle-of-view of the smartphone camera
m : n	Aspect ratio of the video (m≥n)
L	Screen-smartphone distance
αL	Short side length of ▭ ($\alpha = \frac{2n \tan \frac{\theta}{2}}{\sqrt{m^2+n^2}}$)
▭	Captured region
◯	Circle centered on the center of ▭
R	Number of pixels of the captured image corresponding to αL
P	Number of pixels of the captured image corresponding to M
P_{min}	Minimum P required to detect marker

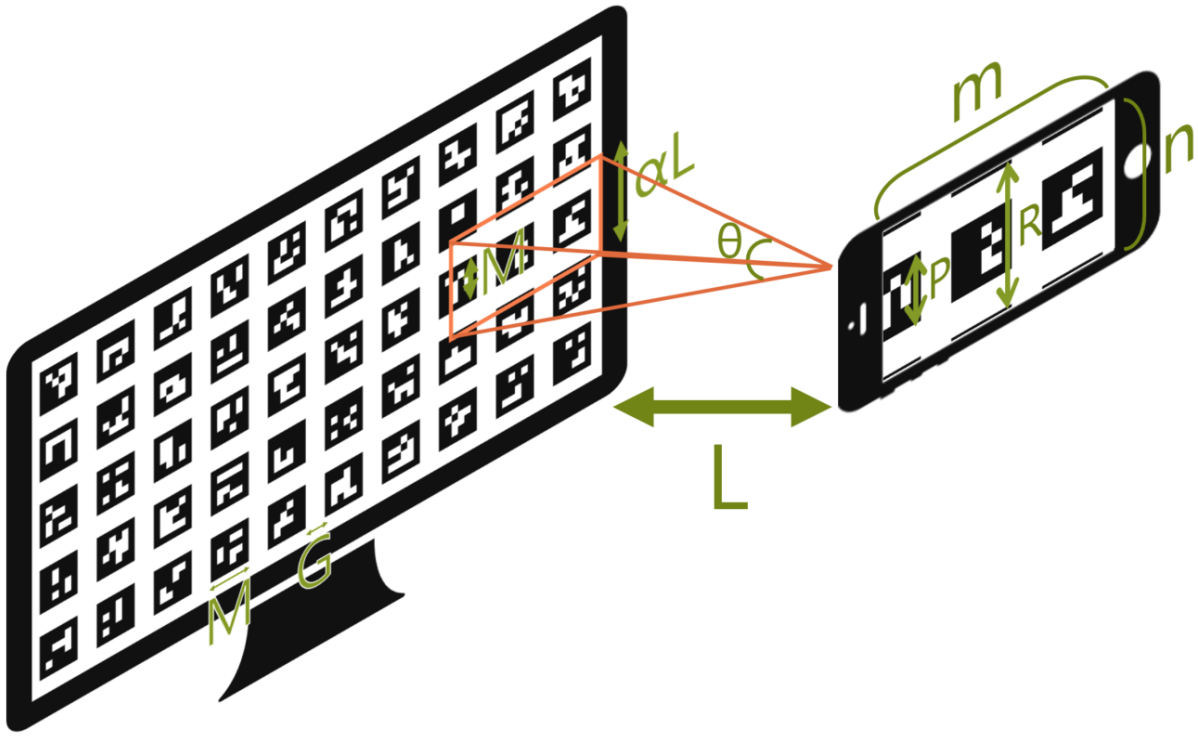


Figure3.3: Variables.

Chapter 4

Experiments

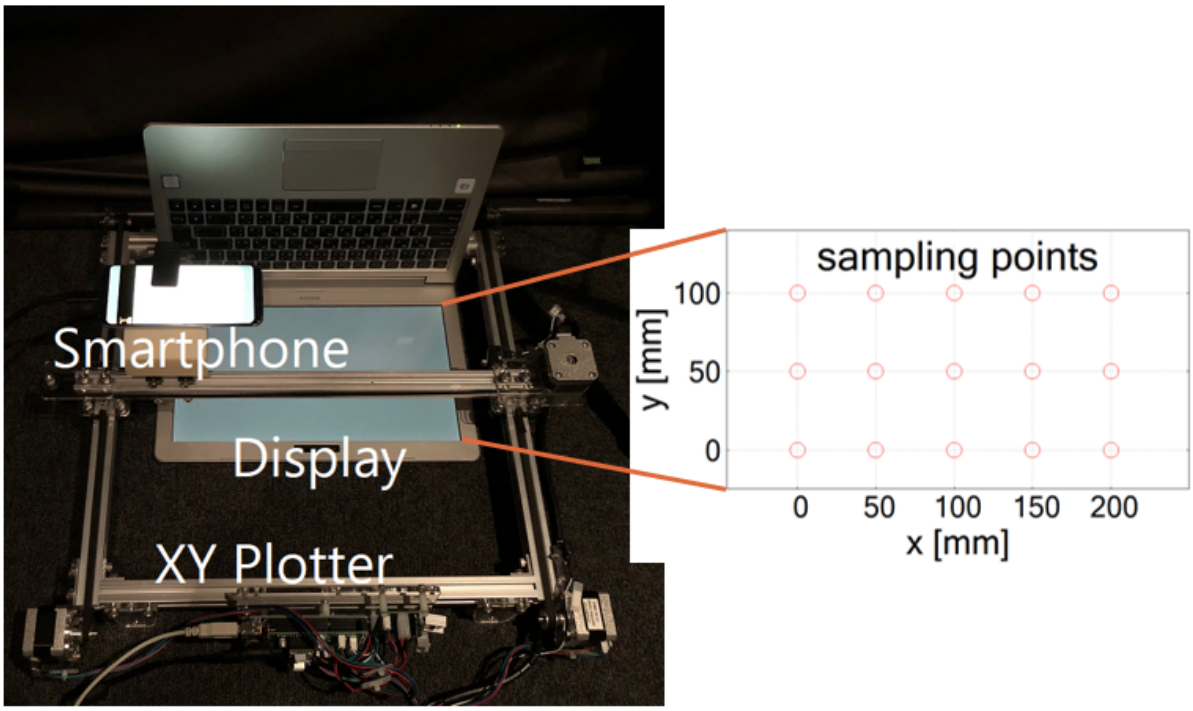


Figure4.1: Setup for Experiment 1.

We conducted four experiments for measuring the display pointing accuracy and the maximum length between the smartphone and screen, and for evaluating the robustness against motion and rotation, respectively. We used a 13.3-inch 1080p laptop (MB-S250XN1-EX3, Mouse Computer) as the screen and the Galaxy S8 (Samsung) as the smartphone ($\theta \approx 70°, m = 16, n = 9, R = 720$ px). The experiments were conducted in an assembled darkroom (ADR-F2, ASONE).

4.1 Experiment-1: Pointing Accuracy

4.1.1 Condition

To avoid error between the ground truth and the actual position of the smartphone, we estimated the relative position from a certain position, where the smartphone was pointed at.

Figure 4.1 shows the experimental setup. We embedded 10×5 AR markers ($M = 120$ px, $G = 60$ px) into a gray single-color image ((R, G, B) = (128, 128, 128)). We used the ArUco [34,35] to generate 4×4 bit markers. The modulated colors were (R, G, B) = (158, 117, 131) and (R, G, B) = (85, 138, 125). Figure 4.2 displays the actual images. We used an XY plotter to move the smartphone in increments of 50 mm, parallel to the edge of the display. The smartphone was positioned 140-mm above the display. We measured at 5×3 points, starting from a certain point on the upper-left of the display. We performed 100 measurements at each point, and calculated

the relative position by subtracting the measured value from the starting point. In addition, to determine the accuracy of pointing using the AR markers themselves, we conducted the same experiment with visible markers, i.e., black ((R, G, B) = (0, 0, 0)) markers on a white ((R, G, B) = (255, 255, 255)) background using the images shown in Figure 4.2.

4.1.2 Result

We defined the error as the distance between the ground truth and the measured value. The results are depicted in Figure 4.3. The median of the error with visible markers was 0.30 mm and the maximum error was 0.80 mm. In contrast, the median of the error with imperceptible markers was 0.29 mm and the maximum error was 0.92 mm. There were no significant differences between the visible and imperceptible markers. It was confirmed that our system has sufficient accuracy for practical applications because the error was within 1 mm.

4.2 Experiment-2: Maximum Distance Between the Smartphone and Screen

4.2.1 Condition

We measured the detection rate at various values of L (distance between the smartphone and screen) and M (marker length), in this experiment. Figure 4.4 shows the experimental setup. We positioned the smartphone to capture the center-part of the display, and moved it perpendicular to the display in increments of 50 mm. We performed 100 measurements at each point and calculated the detection rate. We used the same gray image and modulated colors as in Experiment-1. We used multiple sizes and varied the numbers of markers ($M = 9.24, 13.85, 18.47, 23.09$ mm).

4.2.2 Result

The result of detection rate is shown in Figure 4.5. The starting point is different for each M because the range of L changes with respect to M, as shown in the right side of inequality (3.3). We can estimate the maximum L with respect to M from the result. We plotted the largest L for which the detection rate was higher than 50% for each M (Figure 4.6); line, L_{max}, is their least-squares approximation ($L_{max} \approx 24M$). The value of P_{min} can be calculated from the left-side of inequality (3.3) and the slope of L_{max}, i.e., $P_{min} \approx 43$. Line, L_{min}, is derived from the right-side of inequality (3.3), when $G = 30$ px ($L_{min} \approx 4M + 9.5$). Hence, the values of L between L_{max} and L_{min} are the distances over which our system works. For example, the smartphone captured approximately 5% of the screen at $L = 50$ mm and 100% at $L = 230$ mm, under the conditions of the conducted experiment.

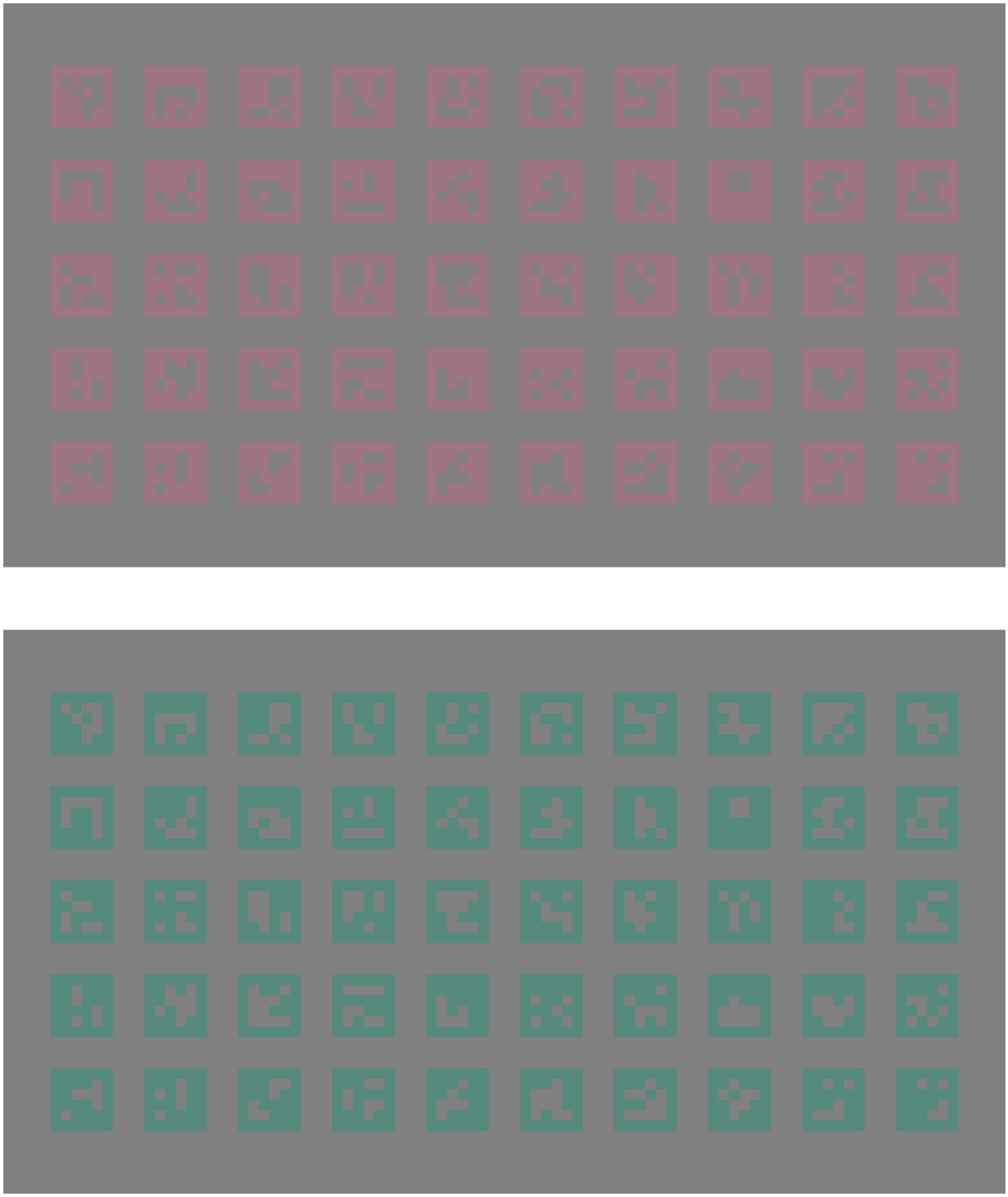


Figure4.2: Images used in Experiment-1; these two images are displayed alternately at 60 Hz.

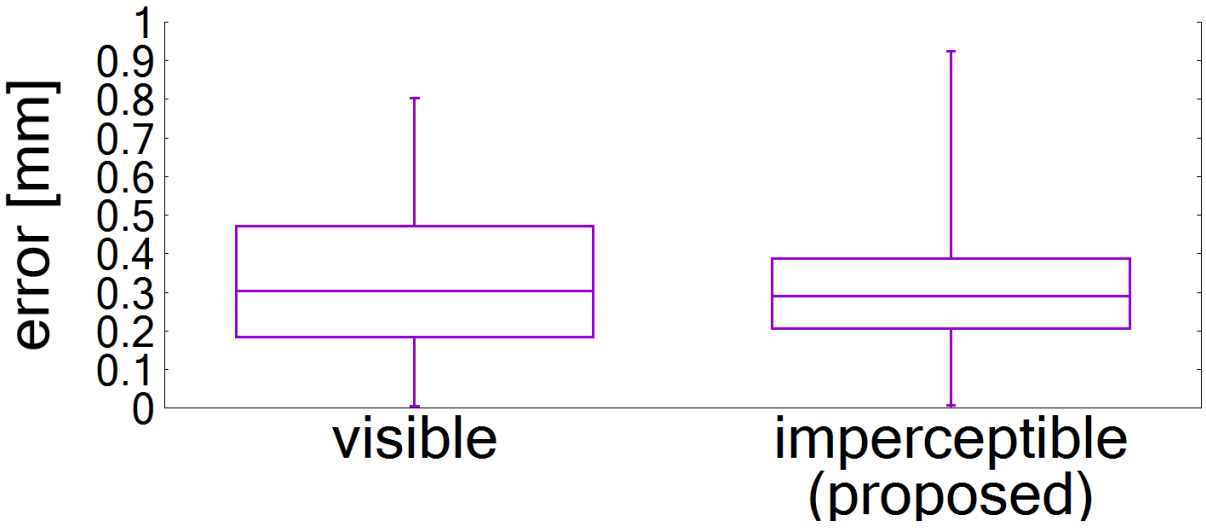


Figure4.3: Error results; there were no significant differences between the visible and imperceptible markers.

4.3 Experiment-3: Detection Rate with a Moving Smartphone

4.3.1 Condition

We measured the detection rate, when the smartphone was moving. Figure 4.7 shows the experimental setup. The smartphone was set on an electrical linear slide (EZS6-D085-AZMAD-1, Oriental Motor) and moved at constant speed, 140 mm above the display . The distance moved from the starting point was measured. We performed 100 measurements at each speed and calculated the detection rate. The same gray image, modulated colors, and embedded markers, as in Experiment-1, were used.

4.3.2 Result

The result is depicted in Figure 4.8. The markers could not be detected at more than 45 mm/s. Figure 4.9 shows the measured values and ground truth at 15 mm/s and 30 mm/s. The ground truth includes the raw data from the linear slide. The measured values were smooth at 15 mm/s and jagged at 30 mm/s. Although our system performance was unsatisfactory when the smartphone was moving, this result implies that our system can withstand camera shake.

4.4 Experiment-4: Detection Rate with a Tilted Smartphone

4.4.1 Condition

We measured the detection rate, when the smartphone was tilted. The smartphone was positioned 90-mm above and parallel to the display, and rotated in increments of 15° along the

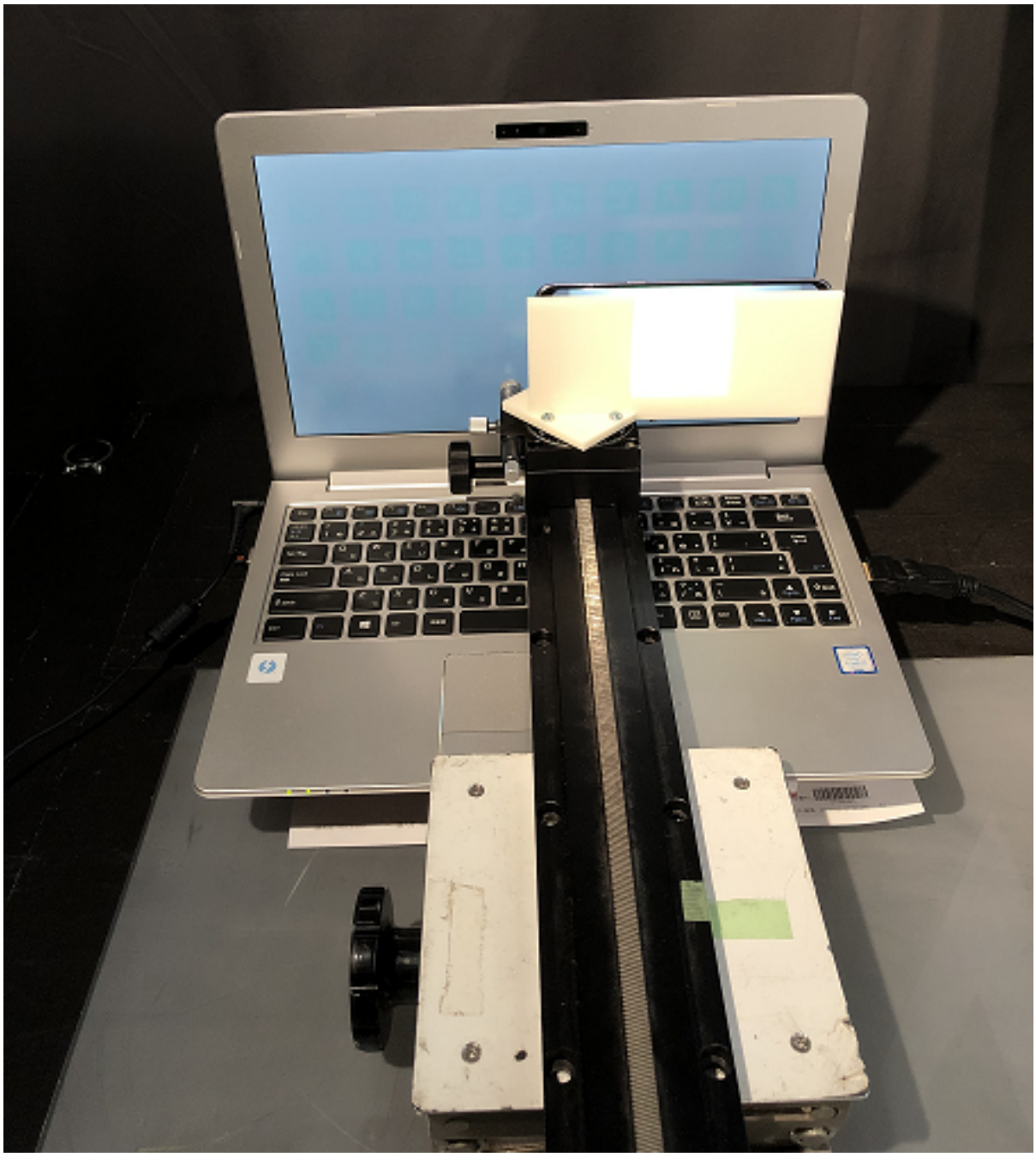


Figure4.4: Setup for Experiment 2.

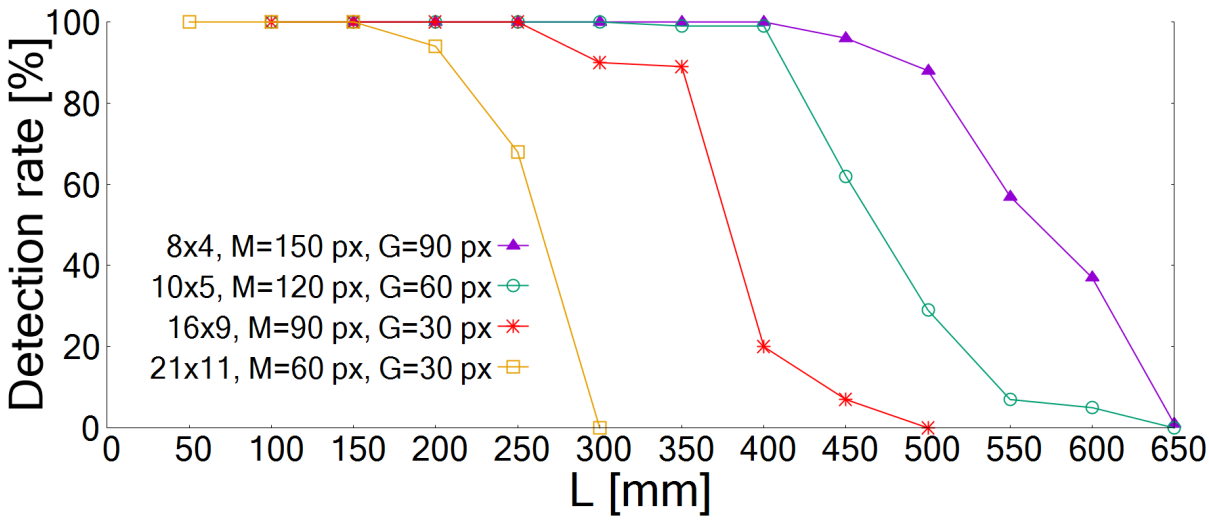


Figure4.5: Detection rates for different M (marker size) and L (distance between the smartphone and screen). Each legend shows the number of markers, M, and G (marker interval).

three axes defined in Figure 4.10, respectively. Figure 4.11– 4.13 shows the experimental setup for each axis. We measured the distance from the pointing position at $0°$ to the pointing position at each angle. Errors within 8 mm were defined as correct detection. We performed 100 measurements at each angle and calculated the detection rate. We used the same gray image, modulated colors, and embedded markers as in Experiment-1.

4.4.2 Result

The result is depicted in Figure 4.14, confirming that our system is robust against rotation.

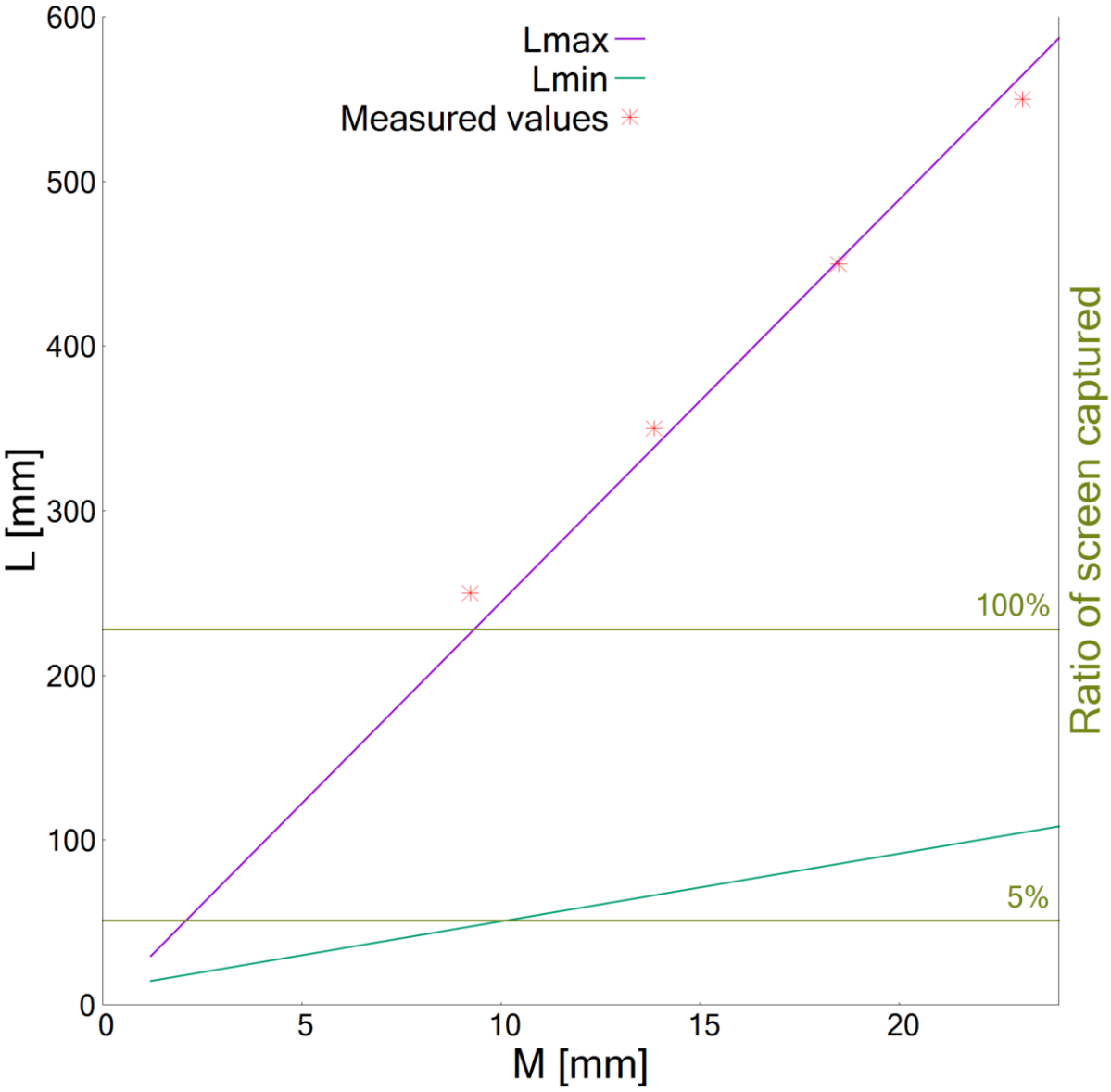


Figure4.6: Range of L according to M. The points indicate the largest L at which the measured detection rate is higher than 50% for each M, L_{max} is their least squares approximation, and L_{min} is a line derived from the right-side of inequality (3.3). Our system can be used in the range between L_{max} and L_{min}. The green horizontal line indicates the ratio of the screen captured by the smartphone from a distance, L.

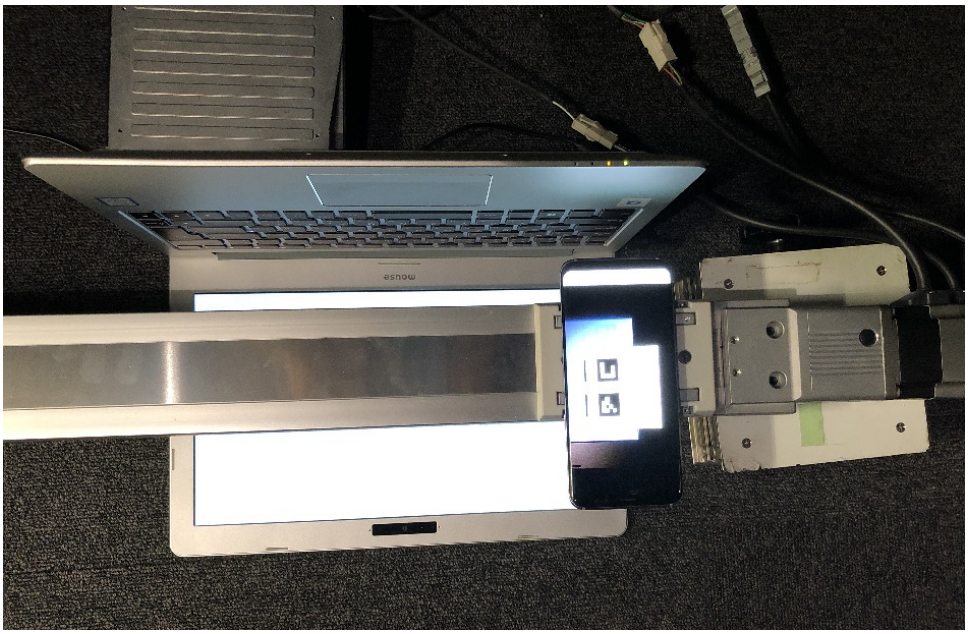


Figure4.7: Setup for Experiment 3.

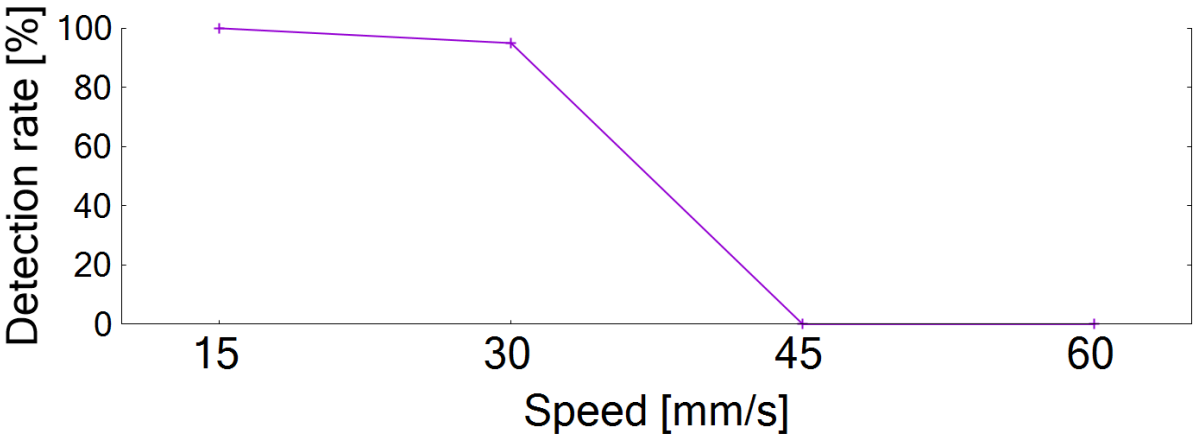


Figure4.8: Detection rates for different smartphone movement speeds.

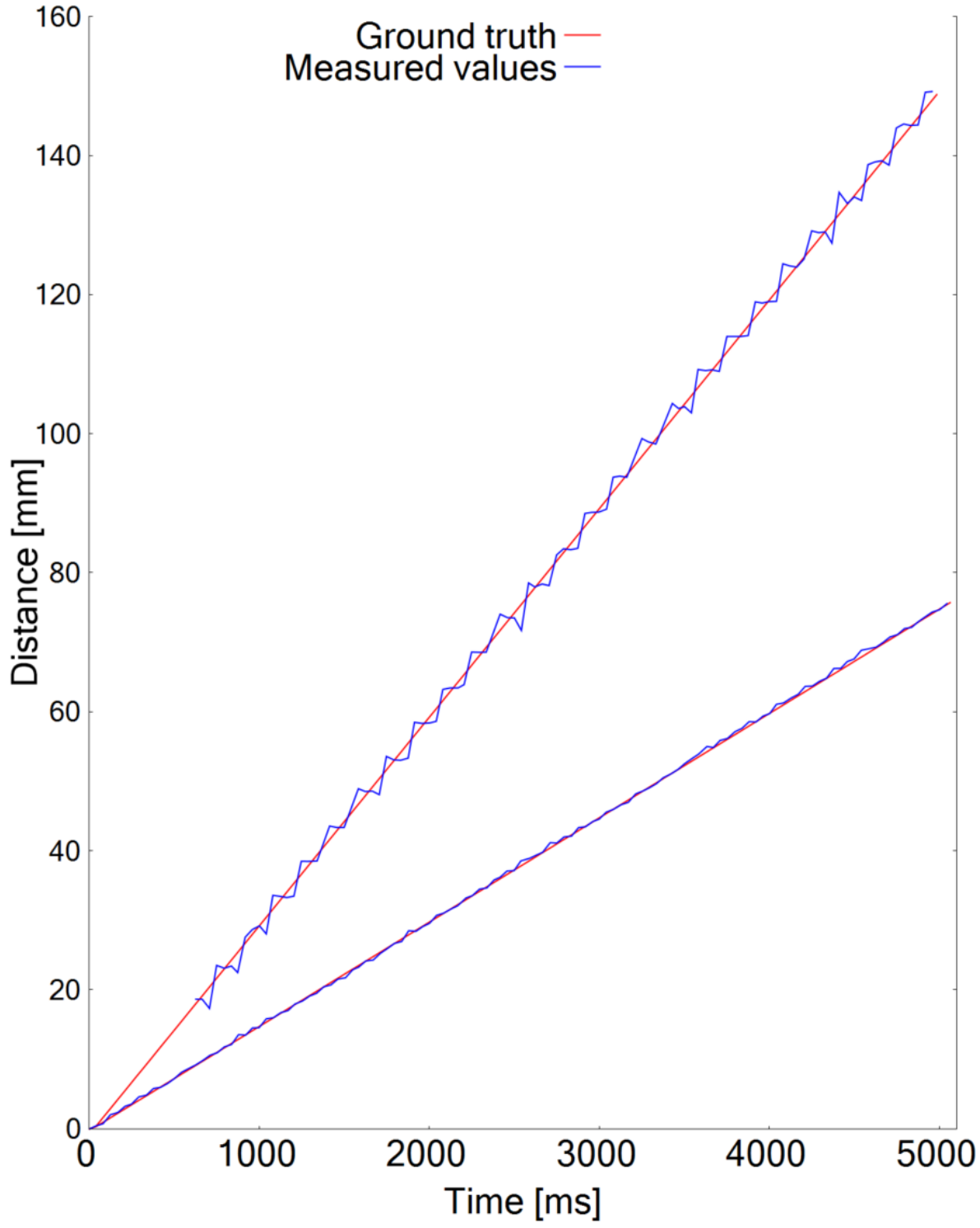


Figure4.9: Measured values for smartphone speeds of 15 and 30 mm/s. The ground truth includes raw data from the linear slide. The markers could not be detected immediately, after the movement commenced at 30 mm/s. This result implies that our system can withstand camera shake.

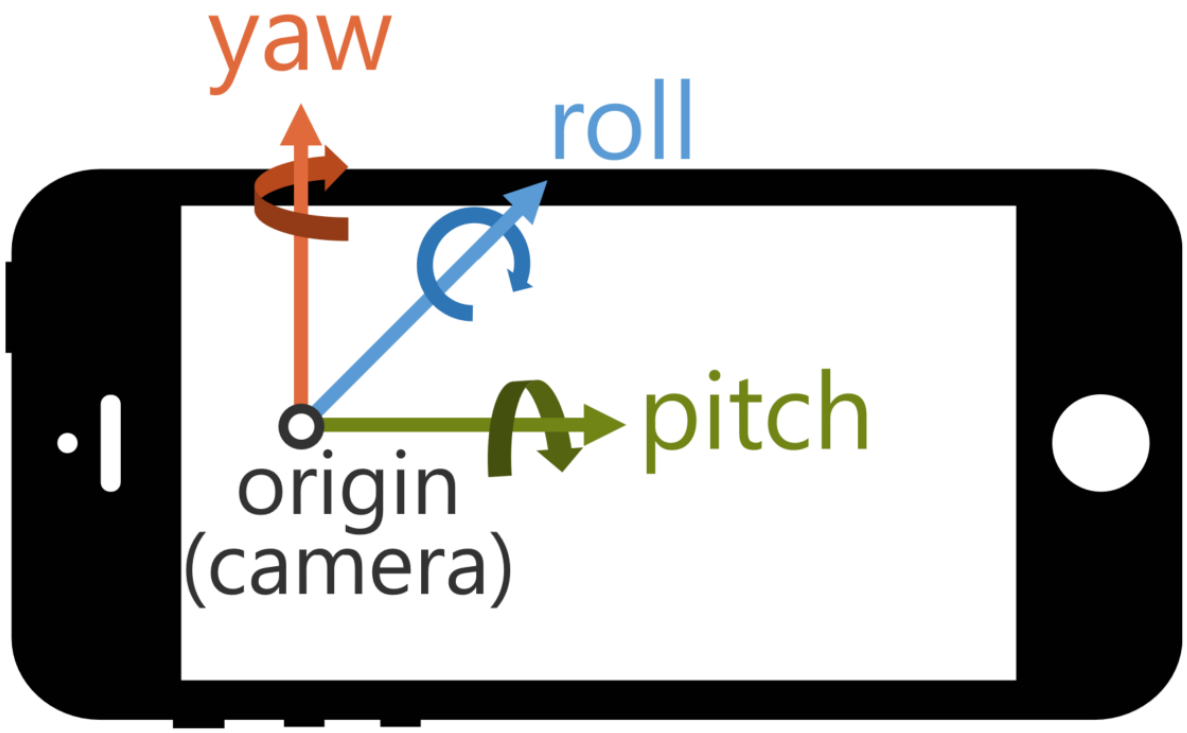


Figure4.10: Definition of the three rotation axes.

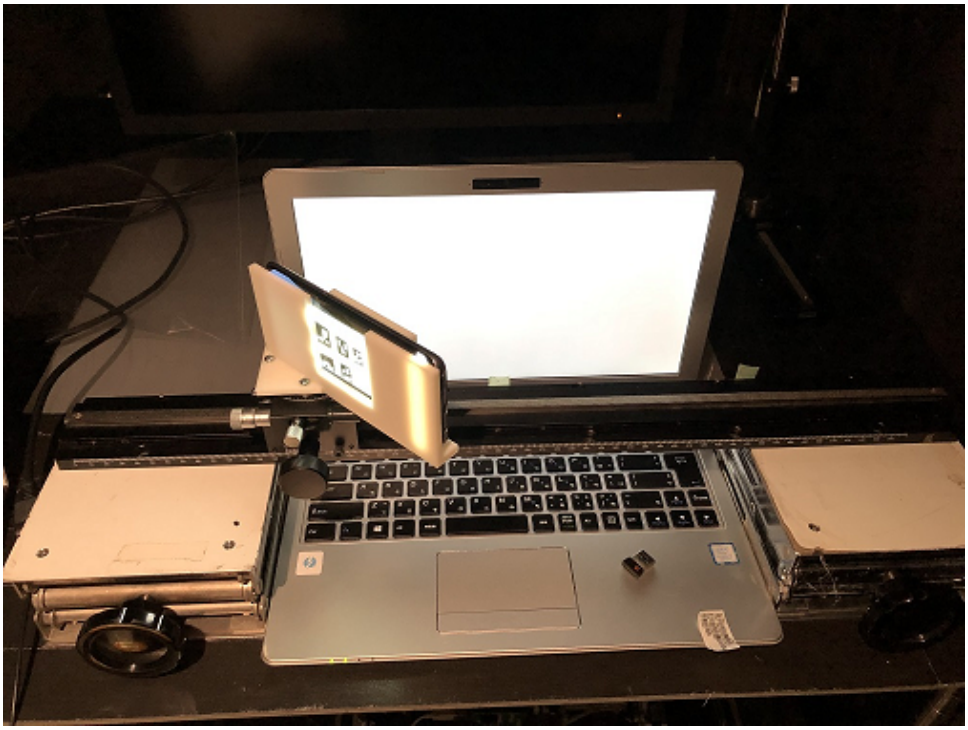


Figure4.11: Setup for Experiment 4 (yaw).

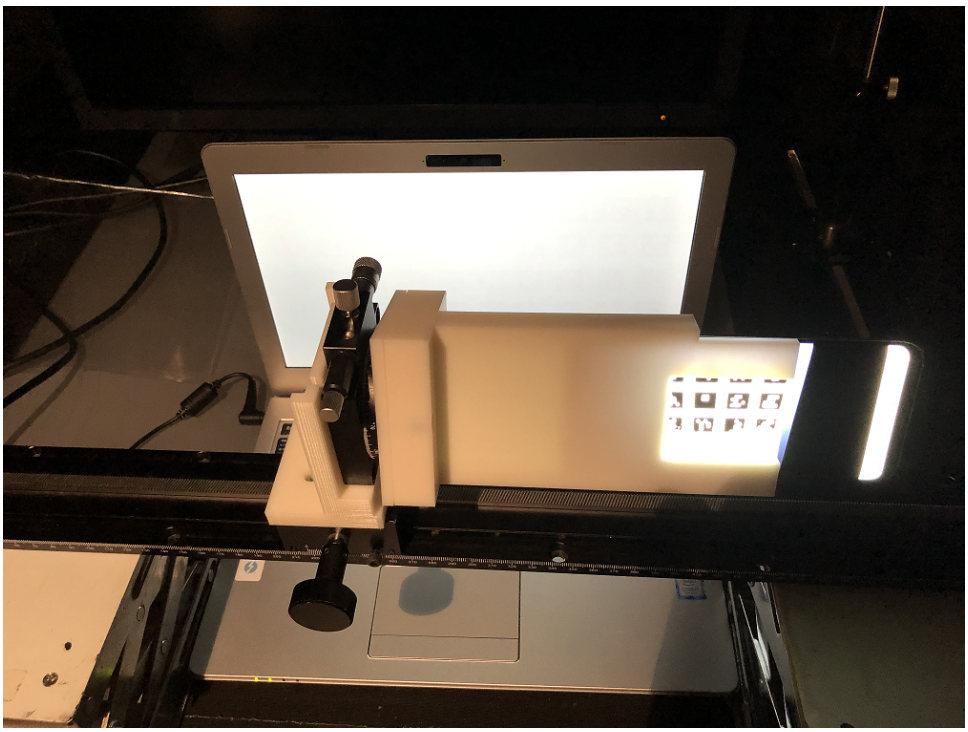


Figure4.12: Setup for Experiment 4 (pitch).

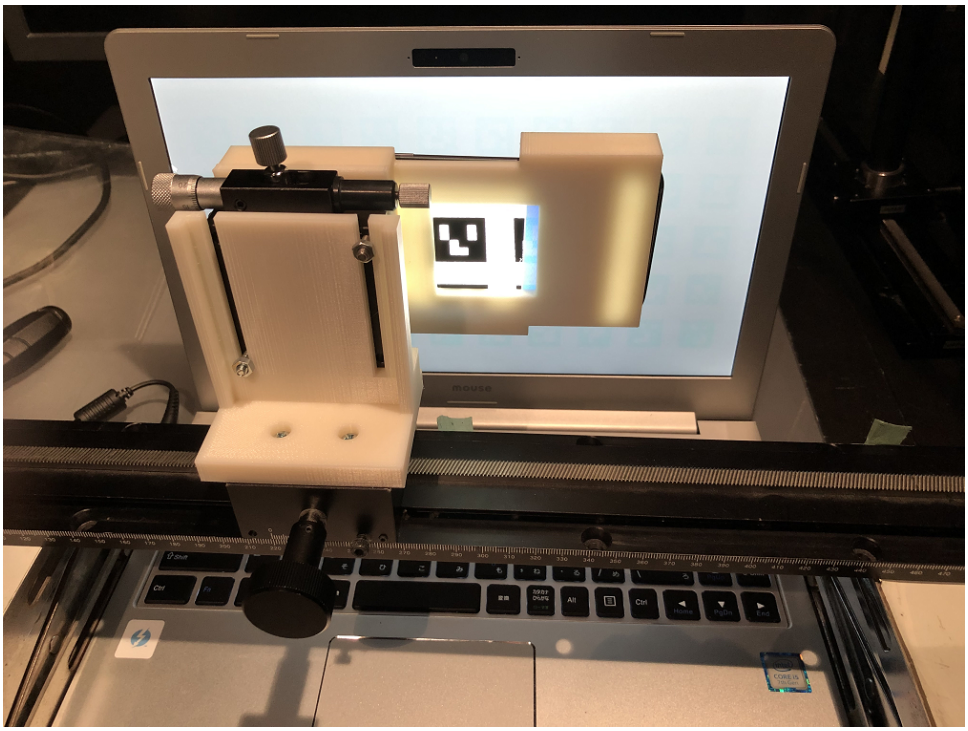


Figure4.13: Setup for Experiment 4 (roll).

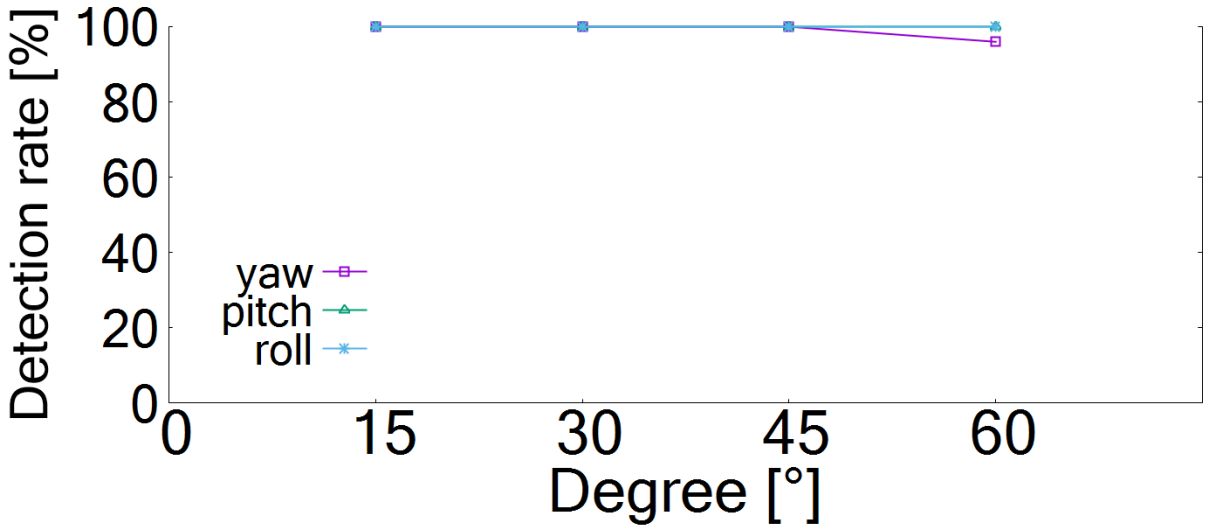


Figure4.14: Detection rates for the three rotation axes. It is confirmed that our system is robust against rotation.

Chapter 5

Applications

We developed an application in which the display screen depicts a floor map and a user acquires information on a region of the map by pointing a smartphone at it (Figure 1.1). The user can either jump to a related webpage or obtain directions to the spot indicated on the map. We confirmed that the AR markers worked for various positions of the hand-held smartphone. This application demonstrates the potential of this method for application in fields such as advertising and indoor navigation.

We also developed an application of the world map. The smartphone calculates longitude and latitude from the pixel position where the camera center is pointing, and then it conducts reverse geocoding to get the country name.

In Figure 5.2 the smartphone is displaying the decoded image and the pixel position at which it is pointing.

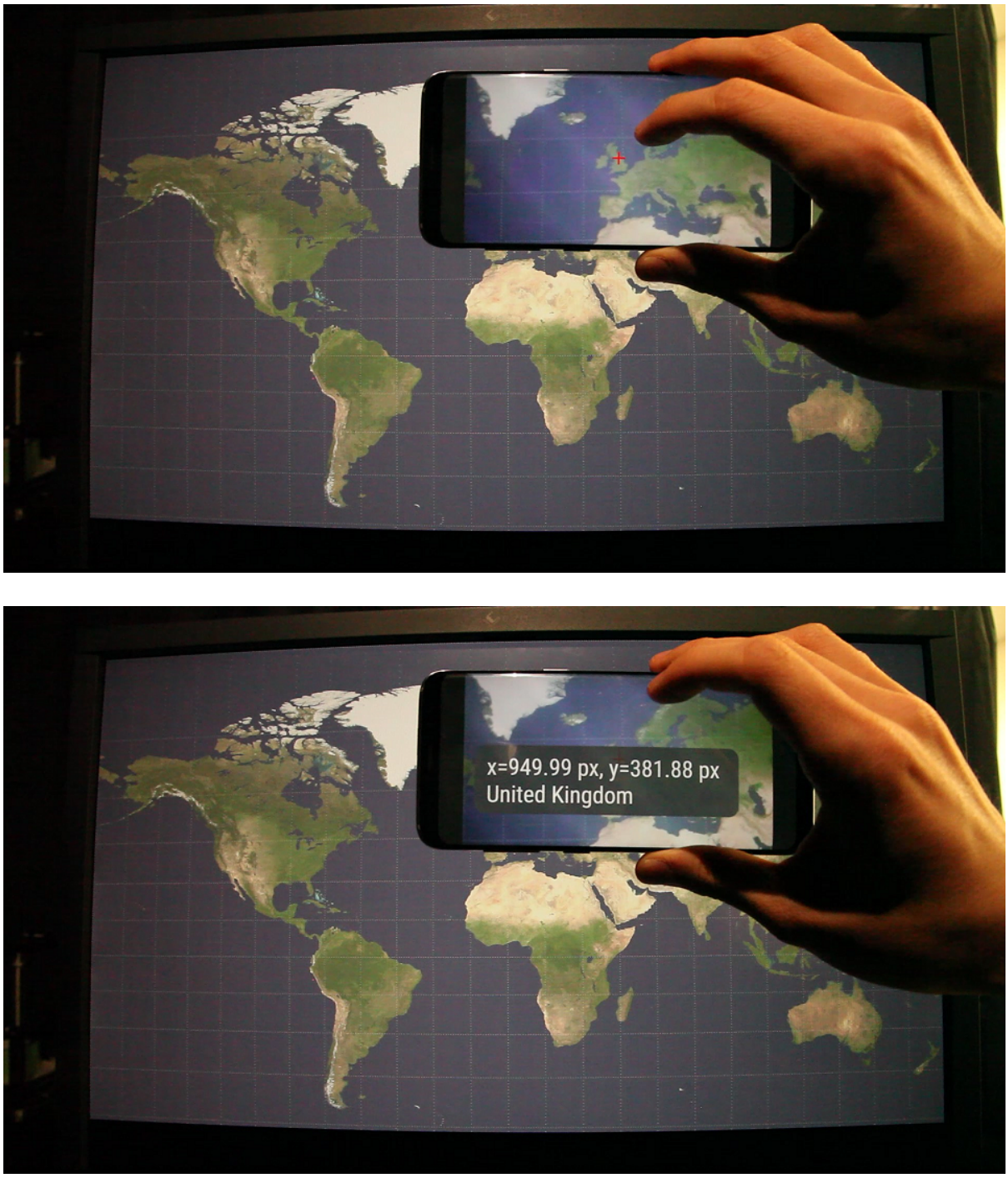


Figure5.1: World map application.

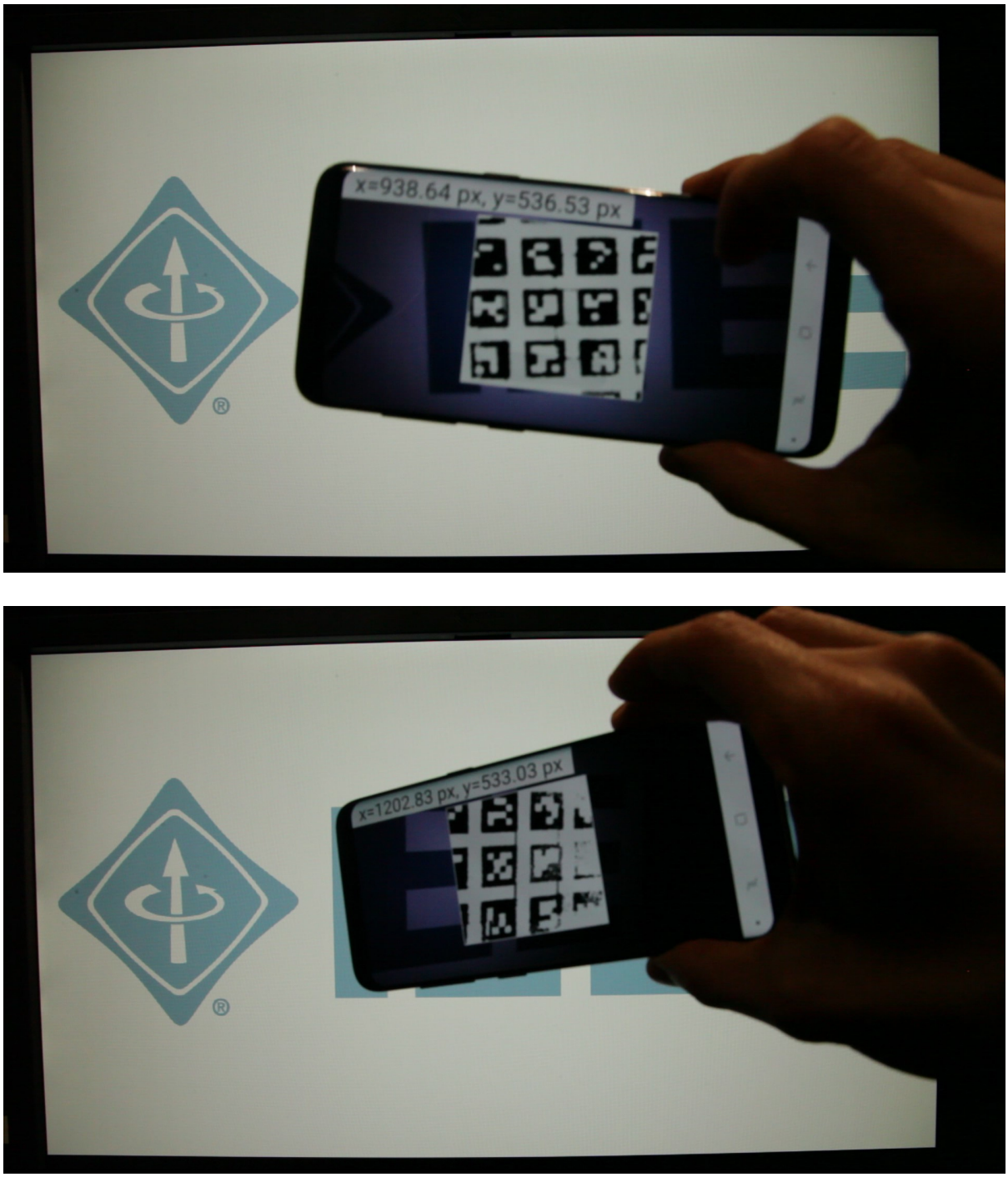


Figure5.2: The smartphone is displaying the decoded image.

Chapter 6

Conclusion

In this study, we proposed an easy-to-install method for localizing a smartphone near a screen, with embedded AR markers through imperceptible color vibration. We demonstrated that our system has sufficient accuracy and robustness against rotation for practical applications, and clarified the relationship between the marker-size and distance-range. Specifically, our system works, even when the smartphone captures only 5% of the screen. In addition, we developed an application to demonstrate our method, and established the potential of the proposed method for application in fields such as advertising and indoor navigation.

A limitation of our method is that our system does not work, when the smartphone is placed on the screen. In future, we aim to develop a method for tracking smartphones that are placed on the display screen. Our system will be more practical, if we can create an embedded marker that can handle a wider distance range.

Acknowledgment

I would first like to express my appreciation to my supervisor Professor Takeshi Naemura for the generous support of the research direction, the environment maintenance, and everything else. I was very fortunate to be a member of his laboratory.

I am also grateful to Assistant Professor Shogo Fukushima and Dr. Takefumi Hiraki, who mainly directed me in research and provided sharp comments whenever I asked. My gratitude also goes to Mr. Satoshi Abe since my research would not have even started without his research. Discussions with these co-workers have been always illuminating and helped me a lot. Last in order but not of importance, I thank all the laboratory members for their cooperation.

Finally and most importantly, special thanks to my friends and family for their moral support and warm encouragements.

December 21st, 2019
Akira Matsumoto

Bibliography

[1] S. Abe. Imperceptible Color Vibration for Screen-Camera Communication via 2D Binary Pattern. *Department of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo, Master Thesis*, 2019.

[2] S. Abe, A. Arami, T. Hiraki, S. Fukushima, and T. Naemura. Imperceptible Color Vibration for Embedding Pixel-by-Pixel Data into LCD Images. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, pages 1464–1470, 2017.

[3] S. Abe, T. Hiraki, S. Fukushima, and T. Naemura. Screen-Camera Communication via Matrix Barcode Utilizing Imperceptible Color Vibration. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings*, UIST '18 Adjunct, pages 166–168, 2018.

[4] M. Baldauf, M. Salo, S. Suette, and P. Fröhlich. Display Pointing: A Qualitative Study on a Recent Screen Pairing Technique for Smartphones. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*, BCS-HCI '13, pages 45:1–45:6, 2013.

[5] D. Baur, S. Boring, and S. Feiner. Virtual Projection: Exploring Optical Projection As a Metaphor for Multi-device Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1693–1702, 2012.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, June 2008.

[7] S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch. Touch Projector: Mobile Interaction Through Video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2287–2296, New York, NY, USA, 2010. ACM.

[8] L.-W. Chan, H.-T. Wu, H.-S. Kao, J.-C. Ko, H.-R. Lin, M. Y. Chen, J. Hsu, and Y.-P. Hung. Enabling Beyond-surface Interactions for Interactive Surface with an Invisible Projection. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 263–272, 2010.

[9] L. Coconu and H.-C. Hege. devEyes: Tangible Devices on Augmented Passive Surfaces.

In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '17, pages 409–411, New York, NY, USA, 2017. ACM.

[10] T. Hao, R. Zhou, and G. Xing. COBRA: Color Barcode Streaming for Smartphone Systems. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 85–98, New York, NY, USA, 2012. ACM.

[11] T. Hiraki, S. Fukushima, and T. Naemura. Phygital Field: an Integrated Field with a Swarm of Physical Robots and Digital Images. *SIGGRAPH ASIA 2016 Emerging Technologies on - SA '16*, pages 1–2.

[12] W. Hu, H. Gu, and Q. Pu. LightSync: Unsynchronized Visual Communication over Screen-camera Links. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*, MobiCom '13, pages 15–26, New York, NY, USA, 2013. ACM.

[13] ISO/IEC. International symbology specification – MaxiCode, 2000.

[14] ISO/IEC. Automatic identification and data capture techniques – Data Matrix bar code symbology specification, 2006.

[15] ISO/IEC. Automatic identification and data capture techniques – QR Code 2005 bar code symbology specification, 2006.

[16] ISO/IEC. Automatic identification and data capture techniques – Aztec Code bar code symbology specification, 2008.

[17] Y. Jiang, K. Zhou, and S. He. Human visual cortex responds to invisible chromatic flicker. *Nature Neuroscience*, 10:657–662, Apr 2007.

[18] K. Jo, M. Gupta, and S. K. Nayar. DisCo: Display-Camera Communication Using Rolling Shutter Sensors. *ACM Transactions on Graphics*, 35(5):1–13, jul 2016.

[19] Y. Kato, N. Fukasawa, and T. Naemura. iPvlc: Pixel-level Visible Light Communication for Smart Mobile Devices. In *ACM SIGGRAPH 2011 Posters*, SIGGRAPH '11, page 45, 2011.

[20] S. Kimura, M. Kitamura, and T. Naemura. EmiTable: A Tabletop Surface Pervaded with Imperceptible Metadata. *Tabletop 2007 - 2nd Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 189–192, 2007.

[21] M. Kojima, M. Sugimoto, A. Nakamura, M. Tomita, H. Nii, and M. Inami. Augmented Coliseum: An Augmented Game Environment with Small Vehicles. In *Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems, TABLETOP'06*, volume 2006, pages 3–8, 2006.

[22] J. Lee, S. Hudson, and P. Dietz. Hybrid Infrared and Visible Light Projection for Location

Tracking. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pages 57–60, 2007.

[23] J. C. Lee, S. E. Hudson, J. W. Summet, and P. H. Dietz. Moveable Interactive Projected Displays Using Projector Based Tracking. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pages 63–72, New York, NY, USA, 2005. ACM.

[24] J. Y. Lee, M. S. Kim, D. W. Seo, S. M. Lee, and J. S. Kim. Smart and space-aware interactions using smartphones in a shared space. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services Companion*, MobileHCI '12, pages 53–58, 2012.

[25] S. Leigh, P. Schoessler, F. Heibeck, P. Maes, and H. Ishii. THAW: Tangible Interaction with See-Through Augmentation for Smartphones on Computer Screens. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '15*, pages 89–96.

[26] T. Li, C. An, X. Xiao, A. T. Campbell, and X. Zhou. Real-time screen-camera communication behind any scene. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '15, pages 197–211, New York, NY, USA, 2015. ACM.

[27] F. J. MacWilliams and N. J. A. Sloane. Pseudo-random sequences and arrays. *Proceedings of the IEEE*, 64(12):1715–1729, Dec 1976.

[28] V. Nguyen, Y. Tang, A. Ashok, M. Gruteser, K. Dana, W. Hu, E. Wengrowski, and N. Mandayam. High-rate flicker-free screen-camera communication with spatially adaptive embedding. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, San Francisco, CA, USA, apr 2016. IEEE.

[29] N. Pears, D. G. Jackson, and P. Olivier. Smart Phone Interaction with Registered Displays. *IEEE Pervasive Computing*, 8(2):14–21, April 2009.

[30] S. D. Perli, N. Ahmed, and D. Katabi. PixNet: Interference-free Wireless Links Using LCD-camera Pairs. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom '10, pages 137–148, New York, NY, USA, 2010. ACM.

[31] R. Raskar, H. Nii, B. deDecker, Y. Hashimoto, J. Summet, D. Moore, Y. Zhao, J. Westhues, P. Dietz, J. Barnwell, S. Nayar, M. Inami, P. Bekaert, M. Noland, V. Branzoi, and E. Bruns. Prakash: Lighting Aware Motion Capture Using Photosensing Markers and Multiplexed Illuminators. *ACM Trans. Graph.*, 26(3), July 2007.

[32] J. Rekimoto and Y. Ayatsuka. CyberCode: Designing Augmented Reality Environments with Visual Tags. In *Proceedings of DARE 2000 on Designing Augmented Reality Environments*, DARE '00, pages 1–10, New York, NY, USA, 2000. ACM.

[33] H. N. Ricciuti and H. Misiak. The Application of the Constant Method in Determining Critical Flicker Frequency (CFF). *The Journal of General Psychology*, 51(2):213–219, 1954.

[34] S. Garrido-Jurado and R. Muñoz-Salinas and F.J. Madrid-Cuevas and M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[35] S. Garrido-Jurado and R. Muñoz-Salinas and F.J. Madrid-Cuevas and R. Medina-Carnicer. Generation of fiducial marker dictionaries using Mixed Integer Linear Programming. *Pattern Recognition*, 51:481–491, 2016.

[36] J. Sanneblad and L. E. Holmquist. Ubiquitous Graphics: Combining Hand-held and Wall-size Displays to Interact with Large Images. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '06, pages 373–377, New York, NY, USA, 2006. ACM.

[37] D. Schmidt, F. Chehimi, E. Rukzio, and H. Gellersen. PhoneTouch: A Technique for Direct Phone Interaction on Surfaces. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 13–16, New York, NY, USA, 2010. ACM.

[38] S. Shi, L. Chen, W. Hu, and M. Gruteser. Reading between lines: high-rate, non-intrusive visual codes within regular videos via ImplicitCode. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '15*, pages 157–168, New York, New York, USA, 2015. ACM Press.

[39] S. Siddhpuria, S. Malacria, M. Nancel, and E. Lank. Pointing at a distance with everyday smart devices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 173:1–173:11, 2018.

[40] M. Sugimoto, G. Kagotani, M. Kojima, H. Nii, A. Nakamura, and M. Inami. Augmented Coliseum: Display-based Computing for Augmented Reality Inspiration Computing Robot. In *ACM SIGGRAPH 2005 Emerging Technologies*, SIGGRAPH '05, 2005.

[41] H. Uchiyama and H. Saito. Random Dot Markers. In *Proceedings of the 2011 IEEE Virtual Reality Conference*, VR '11, pages 35–38, 2011.

[42] 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 *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys '15*, pages 181–195, New York, USA, 2015. ACM Press.

[43] A. Wang, C. Peng, O. Zhang, G. Shen, and B. Zeng. InFrame: Multiflexing Full-Frame Visible Communication Channel for Humans and Devices. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks - HotNets-XIII*, pages 1–7, Los Angeles, CA, USA, 2014. ACM Press.

[44] Q. Wang, M. Zhou, K. Ren, T. Lei, J. Li, and Z. Wang. RainBar: Robust Application-Driven Visual Communication Using Color Barcodes. In *2015 IEEE 35th International Conference on Distributed Computing Systems*, pages 537–546, June 2015.

[45] G. Woo, A. Lippman, and R. Raskar. VRCodes: Unobtrusive and active visual codes for interaction by exploiting rolling shutter. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 59–64, Nov 2012.

[46] R. Xiao, C. Harrison, K. D. Willis, I. Poupyrev, and S. E. Hudson. Lumitrack: Low Cost, High Precision, High Speed Tracking with Projected m-Sequences. In *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13*, pages 3–12.

[47] R. Xiao, S. Hudson, and C. Harrison. CapCam: Enabling Rapid, Ad-Hoc, Position-Tracked Interactions Between Devices. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*, ISS '16, pages 169–178, 2016.

[48] G. Yamamoto, L. Sampaio, T. Taketomi, C. Sandor, H. Kato, and T. Kuroda. Imperceptible On-Screen Markers for Mobile Interaction on Public Large Displays. *IEICE Transactions on Information and Systems*, E100.D(9):2027–2036, 2017.

[49] S. Yamamoto, H. Tanaka, S. Ando, A. Katayama, and K. Tsutsuguchi. Visual SyncAR: Augmented Reality which Synchronizes Video and Overlaid Information. *The Journal of the Institute of Image Electronics Engineers of Japan*, 43(3):397–403, 2014.

[50] L. Yang, J.-M. Normand, and G. Moreau. Robust Random Dot Markers: Towards Augmented Unprepared Maps with Pure Geographic Features. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, VRST '14, pages 45–54, 2014.

[51] M. Yasumoto and T. Teraoka. VISTouch: Dynamic Three-dimensional Connection Between Multiple Mobile Devices. In *Proceedings of the 6th Augmented Human International Conference*, AH '15, pages 89–92, New York, NY, USA, 2015. ACM.

[52] W. Yuan, K. Dana, A. Ashok, M. Gruteser, and N. Mandayam. Dynamic and invisible messaging for visual MIMO. In *2012 IEEE Workshop on the Applications of Computer Vision (WACV)*, pages 345–352, Breckenridge, CO, USA, jan 2012. IEEE.

[53] T. Zhan, W. Li, X. Chen, and S. Lu. Capturing the Shifting Shapes: Enabling Efficient

Screen-Camera Communication with a Pattern-based Dynamic Barcode. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1):52:1–52:25, Mar. 2018.

Publications

[1] 松本 晟, 阿部 知史, 荒見 篤郎, 平木 剛史, 苗村 健: 不可視の色振動を用いた M 系列による映像上の位置情報伝送の基礎検討. 電子情報通信学会総合大会, H-3-5 (2018.3)

[2] 松本 晟, 阿部 知史, 平木 剛史, 福嶋 政期, 苗村 健: 不可視の色振動を用いた AR マーカによるカメラ位置推定の基礎検討. 日本バーチャルリアリティ学会第 23 回大会, 11D-1 (2018.9)

[3] A. Matsumoto, S. Abe, T. Hiraki, S. Fukushima and T. Naemura, ”Imperceptible AR Markers for Near-Screen Mobile Interaction,” In *IEEE Access*, vol. 7, pp. 79927–79933, 2019

.