Image Sensor Communication for Patient ID Recognition Using Mobile Devices

Akira Uchiyama[†1,†2], Takanori Hirao[†3], Hirozumi Yamaguchi[†1,†2], Teruo Higashino[†1,†2]
[†1]Graduate School of Information Science and Technology, Osaka University
[†2]Japan Science Technology and Agency, CREST
[†3]School of Engineering Science, Osaka University
{utiyama,t-hirao,h-yamagu,higashino}@ist.osaka-u.ac.jp

ABSTRACT

Our research group has developed electronic triage tags for measuring vital signs of patients at critical accidents or disaster sites. Although they have wireless networking facilities to transmit the data to a remote server, it is helpful for doctors at the sites to display the vital signs on their visions by augmented reality. For this purpose, we need recognize identification numbers (IDs) of those tags in doctor's vision to obtain the vital signs of the patients from the remote server. In this paper, we propose image sensor communication for ID recognition using embedded cameras in commercial off-the-shelf mobile devices. An LED is attached to a patient and controlled to send the patient ID by blinking patterns. In order for avoiding bit losses and errors due to jitter and capture delays, we derive a necessary and sufficient condition for image capturing intervals. Moreover, we have designed an image sensor communication system using Manchester code for robustness in terms of clock drift. Through experiments using a prototype, we have confirmed the proposed system can recognize more than 94% of 8 bit IDs in 4.5 seconds on average.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

Keywords

Augmented Reality, Image Sensor Communication, Manchester Code

1. INTRODUCTION

Many researchers have proposed applications in wireless networks for various types of objectives. One of such emerging applications is wireless networks for supporting disaster rescue operations[3, 1, 4]. In our project, we have developed an electronic triage system called eTriage[4]. In eTriage, networked electronic triage tags that are attached to patients collect their vital signs and transfer them to a remote server. The tag consists of a small sensor node called SunSPOT and a vital sensor, and thus has limited capability due to hardware limitation. In this system, doctors need access to information about patients. For this purpose, it is helpful to indicate patient information in doctor's vision by using Augmented Reality (AR) as shown in Fig. 1. A doctor holds a camera of a smartphone or a head-mounted display on patients in his/her vision. Then patient IDs are recognized and sent to the server. According to the corresponding patient information returned from the server, the system displays the information on the doctor's vision by AR. In this manner, the doctor can understand the correspondence between patient information and physical patients in his/her vision. To this objective, we need recognize identification numbers (IDs) of patients (tags) in doctor's vision.

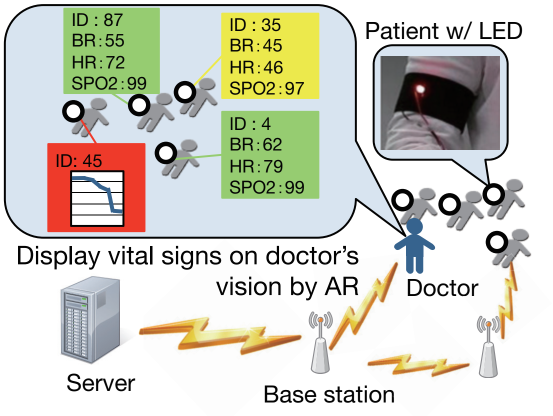

Figure 1: Target Application Overview

There are two categories to recognize objects in AR: markerless methods and marker based methods. Markerless methods[2] rely on natural features of target objects such as edges and corners. For our objective, it is impossible to find differences in natural features between the electronic tags. On the other hand, marker based methods assume fiducial markers attached to objects. Fiducial markers such as square markers[9] are widely used because of its simplicity. However, for object recognition at a long distance, they require large markers which are impractical to stick on objects. For such situations, using Light-Emitting Diodes (LEDs) as fiducial

PhoneSense'12, November 6, 2012, Toronto, ON, Canada.
Copyright 2012 ACM 978-1-4503-1778-8 ...$15.00.

markers is an alternative[10, 5, 6, 7]. The great advantage of LED markers is that they are small and recognizable at a long distance although LEDs require power supply.

In our target environments, electronic triage tags are attached to patients at a disaster site such as a first-aid station. In this situation, we assume tens of patients exist in doctor's vision and recognition of patients more than 10 meters away is required. For this reason, we choose LEDs as fiducial markers. Some existing work has developed dedicated hardware for this purpose. For example, Ref. [5] uses a dedicated CMOS image sensor to capture brightness of each pixel quickly. Picapicamera[7] developed by CASIO exploits LED color information for encoding. Some others combine LED blinking with network communication for synchronization between LEDs and a decoder[10]. However, synchronization causes delays if multiple decoders simultaneously need recognize targets.

Our goal is ID recognition of electronic triage tags more than 10 meters away by using off-the-shelf mobile devices. In our method, LED markers with a 1 cm diameter are used since the electronic tags are small and the capacity of battery is limited. We only rely on brightness information and do not use color information to make the recognition available for low quality cameras. We attach an LED to the electronic tag, and control its blinking pattern by the SunSPOT to transmit its unique ID. Decoders recognize LEDs and track LED blinking patterns for decoding. The challenge is how to handle delays due to timing jitter at transmitters and image capturing delays at decoders. Therefore, we have designed an image sensor communication system using Manchester code[8]. Manchester code enables us to recover clock at decoders since each bit always has one transition. Our contribution is twofold. Firstly, we have derived the necessary and sufficient condition for decoding without bit losses and errors considering clock drift between transmitters and decoders. Secondly, we have confirmed the correctness of the derived condition through experiments using a prototype.

We have implemented the prototype for evaluation. A laptop with an web camera is used for a decoder, and an LED with a diameter of 1 cm is attached to a SunSPOT as a fiducial marker. The experiments have been conducted in bright and dark conditions with decoder's positions at distances of 1, 8, and 15 meter(s) away from the LED. From the results, we have confirmed our system can recognize 94% of 8 bit IDs within 4.5 seconds on average.

2. SYSTEM ARCHITECTURE

2.1 Assumptions

We assume patients are treated at an indoor first-aid station near a disaster site. This indicates a doctor can see tens of patients in a region of which the size ranges from meters squared to ten meters squared. Thus 8 bits are enough to assign unique IDs to each patient in doctor's vision. Even if there are hundreds of patients, it is suffice for obtaining patient information from a server to assign unique IDs to each patient in single doctor's vision. This means we may dynamically change ID assignment using positions and orientations of doctors in order to avoid ID collisions in single doctor's vision. A doctor can obtain information about patients in his/her vision by sending recognized IDs with the current position and orientation to the server. In this paper, ID assignment is out of scope. We focus on ID recognition in single doctor's vision. We also assume doctors stop when they recognize patient IDs and patients do not move.

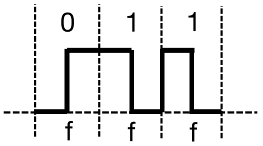

Figure 2: Representation of 011 in Manchester Code

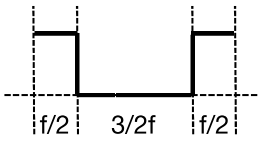

Figure 3: Preamble

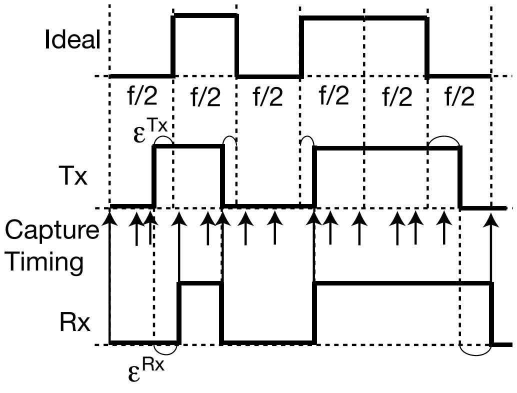

Figure 4: Clock Drift Between Transmitter and Decoder

2.2 Manchester Code and Preamble

As we mentioned earlier, clock drift is not negligible due to delays caused by jitter at transmitters and capturing delays at decoders. Hereafter, we denote the bit period as f. Figure 2 shows representation of data 011 in Manchester code. As we can see, every single bit always has one transition at the middle of each bit period. Because this property enables decoders to recover clock signals, Manchester code is robust in terms of clock drift.

We also need a preamble to indicate the start of ID bits. It is desirable to shorten a preamble as much as possible because a low baud rate is only to be expected due to hardware limitations. Therefore, we define a preamble as shown in Fig. 3. The preamble includes the turned off state having length of $3/2f$ which never appears in Manchester code.

2.3 Clock Drift and Decoding Interval

We now consider details of clock drift. The above two waves in Fig. 4 show an ideal wave and the wave transmitted by the LED. A SunSPOT controls the state of the LED every $f/2$. However, in the transmitted wave, timings of the state transitions become ε^{Tx} earlier or later due to timing jitter. We denote the expected lower and upper bounds of the jitter per $f/2$ as ε^{Tx}_{min} and ε^{Tx}_{max}, respectively.

A decoder reconstructs the transmitted wave by intermittently capturing images. The capture timings are shown by arrows in Fig. 4. The intervals of capture timings are unstable due to hardware limitation. The observed wave at the decoder delays by ε^{Rx} after each state transition as shown in the bottom in Fig. 4. This is because the decoder reconstructs the state transitions of the transmitted wave at the timings when the captured states are opposite to the previous states. For this reason, each period of turned on/off states may be shorter or longer than the corresponding period of the transmitted wave. Let ε^{Rx}_{max} denote the upper bound of the intervals of the capture timings. Then the range of delay ε^{Rx} is $[0, \varepsilon^{Rx}_{max}]$.

The problem caused by the jitter and the capture delays is that we may not be able to distinguish transitions at the middle of bit periods from those at the beginning of bit periods. We let T_c represent the elapsed time since the last decode timing. From the bounds of the jitter and the capture timing delays, the longest period observed by a decoder corresponding to period $f/2$ in ideal waves is

$$f/2 + \varepsilon^{Tx}_{max} + \varepsilon^{Rx}_{max}.$$

Period T_c of observed waves corresponds to period $f/2$ in ideal waves as long as T_c does not exceed the above period. In reverse, when T_c corresponds to $f/2$, T_c does not exceed the above period. Similarly, the shortest period corresponding to f in ideal waves is

$$f + \varepsilon^{Tx}_{min} - \varepsilon^{Rx}_{max}.$$

If period T_c exceeds the above period, T_c of observed waves corresponds to period f in ideal waves, and vice versa.

According to the above conditions, we introduce decoding interval I. I indicates the time period that a decoder has to wait before decoding. If and only if T_c exceeds I, the observed transitions are decoded. Therefore, I has to be greater than the period corresponding to $f/2$ and less than the period corresponding to f. Thus the necessary and sufficient condition for decoding without bit losses/errors is

$$\begin{aligned} &\text{Decode the last transition when } T_c > I \\ \iff\ & f/2 + \varepsilon^{Tx}_{max} + \varepsilon^{Rx}_{max} \leq I \leq f + \varepsilon^{Tx}_{min} - \varepsilon^{Rx}_{max}. \end{aligned} \quad (1)$$

2.4 Light Detection and Tracking from Captured Images

There are two steps for ID recognition: light detection and tracking of blinking patterns. For the light detection, we scan every pixel and extract pixels that have brightness greater than the threshold Y_{TH}. In extracted pixels, we regard neighboring ones as single light l and merge such pixels. Coordinate $p_l = (x_l, y_l)$ and radius r_l of l are derived by computing the smallest circle C_l that includes all pixels composing l. We define p_l and r_l as the center and radius of C_l respectively. In order for noise filtering, we eliminate l if r_l is larger than the threshold r_{TH}, which is empirically set by capturing an image of a turned on LED at a short distance. Finally, the detected lights are added to light list S with their coordinates and radii.

The next step is tracking of the blinking patterns of each light $l \in S$. For this purpose, we identify the same light in successive captured images. Suppose light l is in an image and light m is in the successive image. We regard l and m are the same light if the distance between p_l and p_m is less than l's radius r_l. In this case, we do not add m to S but overwrite l's coordinate and radius by p_m and r_m. Decoding requires tracking the periods of the turned on/off states for each light l. Therefore, elapsed time t_l and current light state s_l for l are also recorded in S. t_l represents the elapsed time since l becomes current state s_l. s_l indicates the state of either turned on or off.

Note that we have to deal with the computation time for the light detection because it scans all pixels. To reduce the overhead, the light detection is conducted once every $f/2$ while the tracking is processed for every captured image.

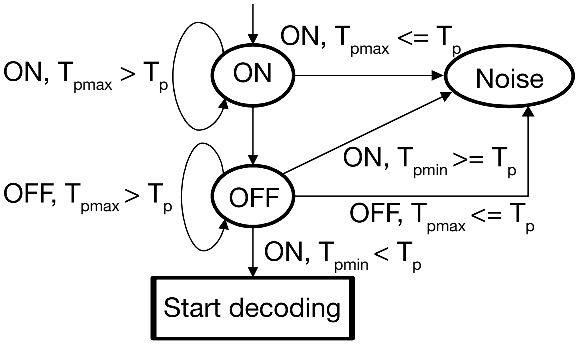

Figure 5: Preamble State Transition Diagram

3. DECODING PATIENT ID

3.1 Preamble Recognition

After l is added to S, a decoder tracks blinking patterns of l to find the preamble. Figure 5 shows the preamble state transition diagram. Here, T_p denotes the period of the previous LED state and is obtained from the elapsed time recorded in the entry of $l \in S$. At first, l starts with the initial ON state in Fig. 5. Then the state becomes OFF when l turns off. During the OFF state, l is kept tracked. Finally, when l is turned on, decoding starts if T_p is greater than the threshold T_{pmin}, which is the minimum length of the OFF state in the preamble. Thus

$$T_{pmin} \leq 3/2f + \varepsilon^{Tx}_{min} - \varepsilon^{Rx}_{max}.$$

Note that we remove noises such as other lights through the above process. If T_p becomes greater than the threshold T_{pmax}, we regard l as a noise and remove it from S. Likewise, the condition that T_{pmax} has to satisfy is

$$T_{pmax} \geq 3/2f + \varepsilon^{Tx}_{max} + \varepsilon^{Rx}_{max}.$$

In addition, if T_p is less than threshold T_{pmin} in the OFF state, we also remove l from S because period T_p of the OFF state is too short compared with the preamble.

3.2 Decoding Data Bits

After preamble recognition, decoders recognize the state transitions at the middle of the bit periods for decoding. As mentioned in Sec. 2.3, we use decoding interval I which indicates the timing when a decoder decodes the observed transition to single data bit. When a state transition is observed, the decoder computes the elapsed time T_c since the last decoding. If T_c exceeds I, we decode the data bit.

Figure 6 shows an example of a decoding process of data bits 001. A decoder holds the current states of decoding for each light. There are four states of decoding as shown in Fig. 7, that are ON/beg, OFF/beg, ON/mid, and OFF/mid. These states represent whether an LED is turned on or off and whether the last observed transition is the beginning or the middle of a bit period. Note that when state transitions to either ON/mid or OFF/mid occur, 0 or 1 is decoded and T_c is reset to 0. When a decoder captures the last $f/2$ state of the preamble as shown in Fig. 6(a), the decoding state is initialized to ON/mid without decoding. The next state is

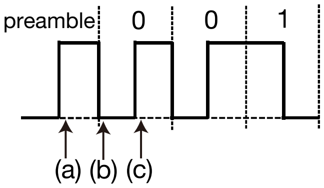

Figure 6: Example of Decoding Process

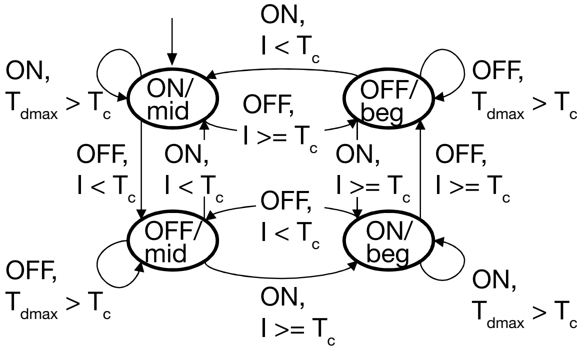

Figure 7: Decoding State Transition Diagram

Figure 8: LED Used in Experiment

OFF/beg since T_c is less than I at the timing of Fig. 6(b). By capturing the next transition at Fig. 6(c), the state transition to ON/mid occurs because T_c exceeds I. The data bit of 0 is decoded at the same time. In the same manner, the following bits are decoded.

In the above decoding processes, we also eliminate noises such as other lights with camera shake and flickers. From the property of Manchester code, the shortest period of the turned on/off state is $f/2$. Therefore, $T_c = f/2+\varepsilon_{min}^{Tx}-\varepsilon_{max}^{Rx}$ is the lower bound of the turned on/off state observed by the decoder. The threshold T_{dmin} is set to a value smaller than this lower bound. If $T_c < T_{dmin}$, the light is filtered as a noise. Likewise, lights that keep turned on/off for a long time are filtered. Threshold $T_{dmax} > f + \varepsilon_{max}^{Tx} + \varepsilon_{max}^{Rx}$ is used for this purpose.

4. EXPERIMENT

4.1 Settings

We have implemented the prototype of our method on a laptop with a webcam by C++ with OpenCV. Figure 8 shows an LED used in the experiment. An LED lens is attached to the LED to focus the light for long distance recognition. The LED is connected to a SunSPOT and controlled by Java running on the SunSPOT. AT91 timer counter is used in the SunSPOT to measure the clock timing as accurately as possible.

Brightness threshold Y_{TH} was empirically set to 240. ε_{max}^{Rx} was $50ms$ in our prototype. Other thresholds were set to values that satisfy each condition.

The experiments were conducted in a large indoor room. A transmitter and a decoder were fixed at each position. We measured two metrics: (i) a recognition success ratio and (ii) an LED detection ratio. The recognition success ratio is defined as

$$\frac{\#\text{ of correct decoding}}{\#\text{ of ID transmission trials}} \times 100(\%).$$

The LED detection ratio is a metric to show the efficiency of LED recognition regardless of the correctness of decoded IDs and is defined as

$$\frac{\#\text{ of wrong decoding} + \#\text{ of correct decoding}}{\#\text{ of ID transmission trials}} \times 100(\%).$$

In the LED detection ratio, decoding with lost bits is not regarded as the wrong decoding. Therefore, the LED detection ratio decreases when bit losses occur. Note that decoding lights except the LED was not observed because thresholds were appropriately set to eliminate noises.

4.2 Effect of Decoding Interval I

To confirm the correctness of the condition (1), we measured the recognition success ratios and the LED detection ratios for different I. Different values of I were set so as to satisfy the condition (1). f was set to $300ms$ or $500ms$. ID was transmitted 40 times for each parameter setting. The distance between the transmitter and the decoder was $8m$.

Figures 9 and 10 show the results in the cases of $f = 300ms$ and $f = 500ms$, respectively. From the results, we can see there are some ranges of I where both the recognition success ratios and the LED detection ratios are 100%. These ranges are covered by the ranges defined by the condition. Overall, the recognition success ratios increase with the increase of I to the lower bound of the condition. However, they decrease when I exceeds the upper bound of the condition due to bit losses. Meanwhile, the LED detection ratios are almost 100% except when I is larger than the upper bound. The reason is that bit errors often occur if I is too short and they do not reduce the LED detection ratios. By contrast, if I is too large, the possibility of bit losses increases. Thus both the recognition success ratios and the LED detection ratios decrease in those cases.

When f is $300ms$, the condition is $200 \geq I \geq 250$. Therefore, the result explains the correctness of the condition because 100% recognition success ratios are observed when the values of I are $225ms$ and $250ms$. The LED detection ratio is still 100% in the case of $I = 200ms$. However, the recognition success ratio decreases to approximately 65% despite satisfying the condition. This result indicates that some data bits were wrongly decoded due to the short decoding interval. This is because the total delay exceeds $\varepsilon_{max}^{Rx}+\varepsilon_{max}^{Tx}$. The above observation is the same in the case of $f = 500ms$ where the condition is $300 \geq I \geq 450$. From the results, it is important that we conservatively set the upper bounds of the delay and the jitter to larger values than measured ones.

4.3 Effect of Distance and Environmental Brightness

For the purpose of seeing the effect of distance, we measured the recognition success ratios at the decoder locations

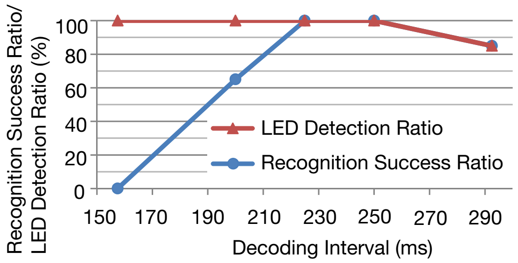

Figure 9: Decoding Interval I vs. Recognition Success Ratio and LED Detection Ratios($f = 300$)

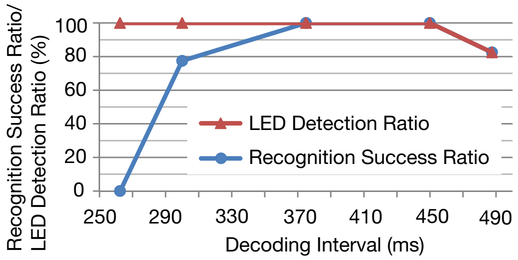

Figure 10: Decoding Interval I vs. Recognition Success Ratio and LED Detection Ratios($f = 500$)

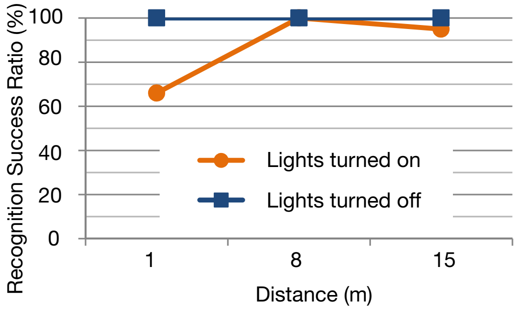

Figure 11: Distance vs. Recognition Success Ratio

1, 8, and 15m away from the transmitter. We used the setting of $I = 225ms$ and $f = 300ms$ for the experiment. In addition, we conducted the experiment in two environments where all room lights were turned on and off.

Figure 11 shows the results. When the lights are turned off, the recognition success ratios are 100% in all cases. The recognition success ratios in the case of the bright environment are little lower but still almost 100%. The only exception is when the distance is 1m where the recognition success ratio is 66%. To investigate the reason, we measured the LED detection ratio, which was 68%. This means that decoding succeeds as long as the LED is detected. Therefore, we conclude that the LED is often regarded as a noise by using noise filtering based on threshold r_{TH}. Nevertheless, our prototype can achieve more than 94% of the recognition success ratios at the distance of 15m. Since $f = 300ms$, the shortest and longest recognition time are approximately 3 and 6 seconds respectively. Thus the average time required to recognize IDs is 4.5 seconds approximately. From the results, we have confirmed our prototype achieves enough performance for our objective.

5. CONCLUSION

In this paper, we proposed image sensor communication for ID recognition using embedded cameras in commercial off-the-shelf mobile devices. In the proposed method, the electronic tags transmit their IDs by LED blinking patterns and decoders continuously capture images to receive those patterns. We have designed our system using Manchester code for robustness in terms of clock drift and derived the necessary and sufficient condition for decoding without bit losses and errors caused by jitter and capture delays. From the results of the experiments using our prototype, we have confirmed the correctness of the derived condition and the prototype can correctly recognize more than 94% of 8 bit IDs in 4.5 seconds on average. Our future work includes handling camera shake and autonomous adjustment of parameters to environmental brightness. We are also planning to implement our prototype on smartphones.

6. REFERENCES

[1] O. Chipara et al. Wiisard: A measurement study of network properties and protocol reliability during an emergency response. In *Proc. of Int'l Conf. on Mobile Systems, Applications, and Services (MobiSys)*, pages 407–420, 2012.

[2] A. Comport, E. Marchand, M. Pressigout, and F. Chaumette. Real-time markerless tracking for augmented reality: The virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):615–628, 2006.

[3] T. Gao et al. The advanced health and disaster aid network: A light-weight wireless medical system for triage. *IEEE Transactions on Biomedical Circuits and Systems*, 1(3):203–216, 2007.

[4] T. Higashino, A. Uchiyama, and K. Yasumoto. eTriage: A wireless communication service platform for advanced rescue operations. In *Proc. of ACM Workshop on Internet of Things and Service Patforms (IoTSP)*, 2011. (invited paper).

[5] N. Matsushita, D. Hihara, T. Ushiro, S. Yoshimura, J. Rekimoto, and Y. Yamamoto. Id cam: a smart camera for scene capturing and id recognition. In *Proc. of Int'l Symp. on Mixed and Augmented Reality*, pages 227–236, 2003.

[6] Y. Oike, M. Ikeda, and K. Asada. A smart image sensor with high-speed feeble id-beacon detection for augmented reality system. In *Proc. of Europian Solid-State Circuits Conf. (ESSCIRC)*, pages 125–128, 2003.

[7] Picapicamera. `http://www.casio.com/news/content/BA2B826C-7962-486A-BC15-67FCFE240F7E/`.

[8] A. S. Tanenbaum. *Computer Networks*. Prentice Hall, 5th edition, 2010.

[9] D. Wagner and D. Schmalstieg. First steps towards handheld augmented reality. In *Proc. of Int'l Conf. Wearable Computers (ISWC)*, pages 127–135, 2003.

[10] G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller, and D. Colucci. High-performance wide-area optical tracking - the hiball tracking system. *Presence: Teleoperators & Virtual Environments*, 10(1):1–21, 2001.