# A Capture-to-Display Delay Measurement System for Visual Communication Applications

Chao Wei*, Haoming Chen**, Mingli Song*, Ming-Ting Sun** and Kevin Lau†
*Zhejiang University, ** University of Washington, † T-Mobile USA
E-mail: *weichaohnu@gmail.com, brooksong@zju.edu.cn, **haominghoward@gmail.com, sun@ee.washington.edu, †Kevin.Lau@T-Mobile.com

***Abstract*— We propose an effective method to measure the capture-to-display delay (CDD) of a visual communication application. The method does not require modifications to the existing system, nor require the encoder and decoder clocks to be synchronized. Furthermore, we propose a solution to solve the multiple overlapped-timestamp problems due to the response time of the display and the exposure time of the camera. We implemented the method in software to measure the capture-to-display delay of a cellphone video chat application over various types of networks. Experiments confirmed the effectiveness of our proposed methods.**

## I. INTRODUCTION

End-to-end delay is an important concern for two-way visual communication applications. Video codec manufacturers need to measure the end-to-end delay of a video codec system in order to develop low-delay video codecs. Network service providers need to make sure the end-to-end delay of a visual communication application is within the application requirement. A simple and general tool for measuring the end-to-end delay of a visual communication system is invaluable for applications related to two-way visual communications.

Figure 1 shows an example of a mobile video chat system. Video captured by the device camera is compressed by a video encoder. The encoder usually contains an encoder buffer to smooth the video bit-rate as described in [1]. The video bit-stream is then packetized and transmitted over the network. At the decoder side, the video is decoded and displayed. The decoder usually contains a decoder buffer to smooth out the network jitter and to buffer the bit-stream before the video decoding. The encoder and decoder buffers can result in a relatively long delay. The end-to-end delay in this example is the latency from frame capturing at the encoder side to the frame display at the decoder side, which we call capture-to-display delay (CDD), including the whole chain of video encoding, encoder buffering, packetization, network transmission, decoder buffering, and video decoding.

The traditional way to measure the latency is by using time stamps. A time stamp is a code representing the global time. It can be generated by a counter from a network clock commonly available to both the encoder and the decoder. To measure the time delay between two points A and B, a time stamp is inserted at Point A, retrieved at the Point B, and compared to the global time at Point B. For example, for the visual communication system shown in Fig. 1, to measure the end-to-end delay of the network part, time stamps are generated from the network clock and inserted at the network interface point in the encoder side. These time stamps are retrieved at the network interface point at the decoder side to compare to the global time. Similarly, to measure the CDD, we can insert timestamps at the video capture point, and observe the timestamps relative to the global time at the display point. As long as a network clock is available or the encoder clock and the decoder clock are synchronized, the delay can be calculated. However, in order to do this, we need to be able to modify the hardware or software to insert the timestamps, and retrieve the timestamps at the desired points. In many situations including our application scenario, cellphone video codecs are implemented in hardware and software by the developers. Thus, we cannot access the inside of a cellphone to insert or retrieve the time stamps. Also, usually the encoder clock and the decoder clock are not synchronized. These make the measuring of the CDD particularly challenging.

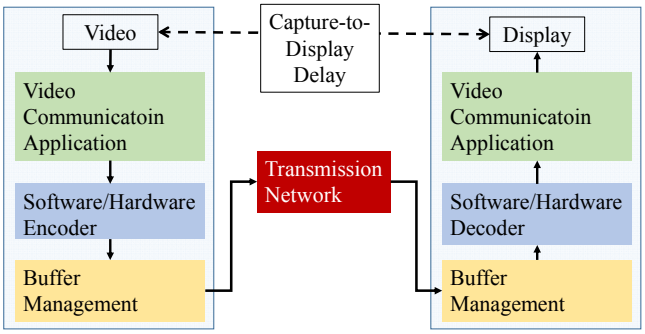


Fig. 1. An end-to-end visual communication system.

Boyaci et al. [2] presented a tool to measure the CDD of a video chat application which does not need any changes in video application software or need specialized hardware. Their approach adds timestamps represented in barcodes at the encoder side. After the video is decoded at the decoder side, the barcodes are recognized and compared to the time at the decoder side. However, this method still requires the development of new software inside the encoder and decoder machines in order to insert and recognize the barcodes and compare the recognized timestamps to the system time. Moreover, the encoder clock and the decoder clock need to be synchronized. In our case, since we cannot access the video

encoder and decoder systems, nor can we guarantee that the encoder clock and the decoder clock in the cellphones on different sides are synchronized, this scheme is not feasible for our situation. To overcome the aforementioned problems, we propose a method that can be used to measure the CDD of any visual communication applications, which does not require modification to the application source code and access to the system. It also does not require the encoder and decoder clocks to be synchronized. It is based on the simultaneous recognitions and comparisons of visual patterns of both timestamps captured at the encoder side and displayed at the decoder side. Our contributions include: (1) propose a new CDD measurement method that does not require modification to the visual communication system nor synchronize the encoder and decoder clocks, and (2) a solution to the multiple-overlapped-timestamp problem encountered using this approach.

The organization of the rest of this paper is as follows. In Section 2, we discuss our approach of measuring the CDD. In Section 3, we discuss the problem of multiple overlapped timestamps and our proposed solution. In Section 4, we present experimental results to show the effectiveness of our proposed methods. Section 5 concludes the paper.

## II. Delay Measurement System Design

### A. The CDD Measurement System

The proposed CDD measurement system is shown in Figure 2. Timestamps representing a stopwatch with millisecond precision are displayed on the screen of an external PC. Cellphone1 and Cellphone2 make the video chat connection. The timestamps shown on the screen of the PC is captured by the Cellphone1 camera. The video of Cellphone1 is encoded and transmitted to Cellphone2. The decoded video is displayed on the screen of Cellphone2. A digital video camera records the timestamps on both the screens of the PC and Cellphone2. The delay measurement system in the PC receives the video from the digital video camera, and performs pattern recognitions for the two timestamp patterns in the video frame. The value of the difference between the two timestamps is the CDD. Note that this system is intended to be used in research for investigating the end-to-end delay. In the application scenario, the network provider has the control on routing the packets or using a network simulator to simulate the network. So the sender/receiver/PC have to be at the same site is not a serious limitation of the proposed approach.

The system is implemented in an external PC which includes two applications: CDD-T and CDD-R. The CDD-T application generates and displays timestamps on the PC screens every M milli-seconds (in the experiments, we set M=30). The CDD-R application receives the captured video from the digital video camera, performs timestamp pattern recognitions, and calculates and displays the CDD of every frame. At the end of a measurement session, the CDD-R application will also output the delay statistics such as the minimum, maximum, and mean CDD and the stand deviation.

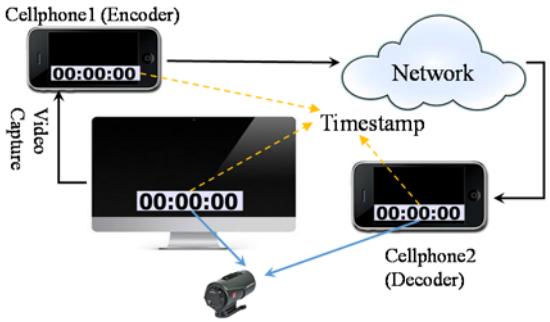


Fig. 2. The proposed approach to measure the CDD.

### B. Timestamps

The timestamps can be represented in different patterns. Three example patterns we investigated to encode the timestamps are shown in Figure 3. Figure 3(a) represents a timestamp in digits and Figure 3(b) represents a timestamp in a QR code [3]. The digit timestamps can be read by the user to get a quick read of the CDD. There is public domain software that can recognize the digit and QR-code timestamps. The QR code is not human readable. However, the fast readability and greater storage capacity compared to the standard barcodes make the QR code increasingly popular. For error-resiliency, Boyaci et al. [2] used an EAN-8 barcode because of its checksum mechanism. The EAN-8 barcode with a checksum can allow to detect whether the timestamp is contaminated, and refuse to read the barcode if it is damaged or distorted. The QR code not only has the checksum mechanism but also the error correction capability. Even if a QR code has some local breakage, it still can restore the original information. Furthermore, the QR code is designed with open standards. Figure 3(c) is a set of special visual patterns we designed to represent the 10 digits. It is still human readable, and due to the simplicity of the pattern, it allows us to see the extent of the multiple overlapped timestamp problem as will be discussed in details in the next section.

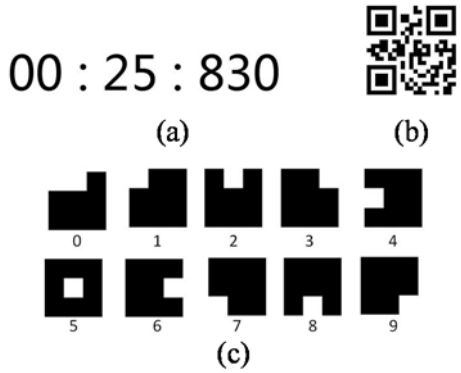


Fig. 3. Examples of timestamp formats (a) digits, (b) QR code and (c) special visual patterns.

## III. Problem of Multiple Overlapped Timestamps And Solution

Although the concept of the above proposed approach looks simple, it has a problem of multiple overlapped timestamps due to the limited response time [4] of the PC and cellphone screens, and the limited camera shutter speed.

### *A. Problems of Multiple Overlapped Timestamps*

When we use a camera to capture the timestamps, the camera needs some time to expose a frame. The exposure process is related to the camera's aperture and shutter speed [5]. The Liquid Crystal Display (LCD) refresh rate [6] is the number of times per second in which the display draws the data it is being given. Due to the limited response time of the display, when a new pattern is displayed, the old pattern may not have completely disappeared. This results in multiple overlapped timestamps on the display as shown in Figure 4. Using public domain software to recognize the multiple overlapped timestamp patterns causes serious errors. We cannot just discard the blurred timestamp frames, because only about 50% of the frames are clear in our experiments. Here, a clear frame means that the timestamps on both the PC and the Cellphone2 screens are clearly recognizable.

To investigate the extent of the multiple overlapped timestamps, we use the visual patterns we developed for representing the digits as shown in Figure 4(c). Since the bright squares in the patterns appear at different locations for different digits, we can easily see how many timestamps are overlapped by counting the numbers of brighter squares in the area representing a digit. In our experiments, we found the number of overlapped timestamps could be as high as four in some occasions.

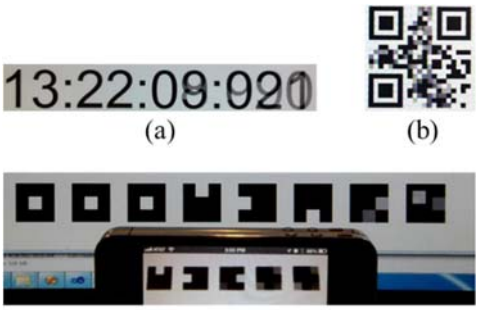


Fig. 4. Multiple overlapped timestamps: (a) with digits, (b) with QR code, (c) with special visual patterns.

### *B. Solution to the multiple overlapped timestamp problem*

We solve the problem with space diversity, i.e., displaying consecutive timestamps at different locations so that they do not overlap before they disappear. The number of space diversity depends on the maximum possible number of overlapped timestamps. Since the maximum number of overlapped timestamps is four in our simulations, we display four consecutive timestamps at four different columns. A video frame showing the QR codes with a space diversity of four is shown in Fig. 5.


Fig. 5. A video frame showing the QR codes with a space diversity of four.

We use bssGenerator [7], a commercial QR code generator Demo SDK, to encode the system time into a QR-code timestamp. With the space diversity of four, the multiple overlapped timestamp problem is solved, but it could result in four faded QR code patterns. In the QR-code recognition, CDD-R first carries out the binarization for every QR-code pattern. Our thresholding method used in the binarization is based on the Otsu algorithm. The details for the Otsu algorithm can be found in [8]. After binarization, the QR code timestamp appearance is much enhanced. Experimental results show that the thresholding in the binarization contributes to improve the recognition accuracy significantly. The binary image is read using ZXing [9], an open source, multi-format 1D/2D barcode reader. The reading accuracy of the QR code shown on the LCD monitor reaches 99.7% and the one on the Cellphone2 screen is above 90% in the experiments (See Section 4). Given four columns of QR codes, CDD-R recognizes all of them first. The columns without QR code observed are removed. The timestamp with the latest time value is chosen as the time of the current video frame. When no recognition of the QR code is successful, the system marks its time as "Invalid". The system performs a correction process for the invalid timestamps after all frames have been recognized. Since the timestamp refresh time is 30ms, the difference between any two timestamps should be a multiple of 30 ms. We can interpolate the invalid time from the successful recognition results based on this property. Since the rate of invalid time is less than 10%, and most of the invalid time is isolated, the delay correction process is quite effective.

## IV. Simulation Results

In our experiments, we pay special attention to the accuracy, precision, and recognition time of different methods. The precision is the percentage of the number of correct recognitions to the total number of successful reads. A successful read does not mean a correct recognition because of possible recognition errors. The accuracy is the rate of the number of frames recognized correctly to the number of all frames.

In the beginning of our experiments, we tested the timestamps in digits without spatial diversity as in Section III.B. The accuracy and precision for the timestamps on the PC screen are 95.5% and 99.4%, respectively which is good. However, the accuracy and precision for the timestamps on the Cellphone2 screen are only 17.2% and 46.9%,

respectively. It means that among 25 frames, only about 12 frames are read successfully and only 4 frames are recognized correctly. The poor result is due to the multiple overlapped-timestamp problem. We also investigated the recognition time. Without applying the proposed diversity scheme, it takes 30.7 ms/frame. The computer we used has a 3.1 GHz Intel Core i3-2100 CPU, 4GB of RAM, and a NVIDIA GeForce 7600 GT graphics card. The machine runs the Windows 7 operating system.

We then experiment the QR code and the visual patterns in Fig.3 (c) with a spatial diversity of four. Table I shows the results. These results confirm the effectiveness of our proposed strategy. The precision of the QR code reaches 100% for both the PC and Cellphone2 screens due to its checksum mechanism and error correction capability. The results using the visual patterns in Fig. 3 (c) is also good.

Table I also shows the read time of a frame with a resolution 720×1280. The read time includes the recognition time and the selection time. The system needs to read two groups of timestamps (one on the PC screen and the other on the Cellphone2 screen) and select the latest time (among four columns due to the diversity of four) for each group separately. The system based on the QR-code takes 69.3 ms to process a frame. The system based on the visual patterns in Fig. 3(c) takes 38.7 ms/frame. The system based on digits in Fig.3 (a) takes 22.0 ms/frame. It should be noted that we have not tried to optimize the speed of the code. With optimization, the speed should be able to be further improved. The QR-code takes more time due to its error correction function.

We have fully implemented the delay measurement system and successfully measured the CDD of many two-way visual communication applications over various networks. Figure 6 shows the performance of using FaceTime over the UW WiFi network when a video session is established through a stable wireless connection. To show the resultant figure clearly, we choose a video consisting of 400 frames to conduct the experiment. From the figure, we can clearly see that the delays of all frames in this example are relatively stable. The minimal delay is 210 ms, the maximal delay is 360 ms, the mean delay is 284 ms, and the standard deviation is about 26 ms.

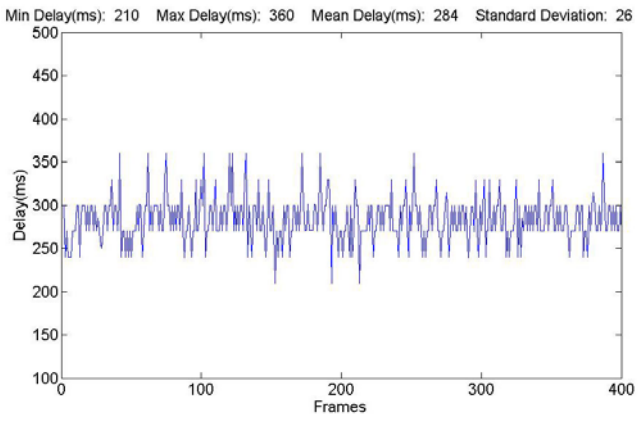


Fig. 6 CDD of a video char session with the QR code.

TABLE I ACCURACY, PRECISION, AND RECOGNITION TIME OF TIMESTAMPS RECOGNITION FOR ONE FRAME (RESOLUTION 720×1280)

| | | QR Code | Visual Patterns | Digits |
|---|---|---|---|---|
| Accuracy (%) | TimeOnPC | 99.7 | 98.8 | 97.1 |
| | TimeOn Phone | 90.8 | 95.2 | 60.5 |
| Precision (%) | TimeOnPC | 100 | 98.9 | 99.5 |
| | TimeOn Phone | 100 | 95.4 | 83.4 |
| Read Time (ms/frame) | | 69.3 | 38.7 | 22.0 |

* The accuracy is the rate of the number of frames read correctly to the number of all frames. The precision is the percentage of the number of correct recognitions to the total number of successful reads.

## V. CONCLUSIONS

We developed an approach to measure the capture-to-display delay of visual communication applications. The approach does not require modifications to any video application source code nor access to the internal of the existing system. Also, it does not require the encoder and decoder clocks to be synchronized. The method is universal so that it can be used to measure the capture-to-display delay of any visual communication applications. It has been successfully implemented in software. We have also proposed a solution with spatial diversity to solve the multiple-overlapped-timestamp problem associated with the proposed approach. Experimental results confirm the effectiveness of the proposed approach.

## VI. ACKNOWLEDGEMENT

The authors would like to thank the support of Chao Wei and Mingli Song from the National High Technology Research and Development Program of China (863 Program) under grant No. 2013AA040601. The views presented are as individuals and do not necessarily reflect any position of T-Mobile USA.

## REFERENCES

[1] A. R. Reibman and B. G. Haskell, "Constraints on variable bit-rate video for ATM networks," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 2, pp. 361-372, Dec, 1992.

[2] O. Boyaci, A. Forte, S. Abdul Baset, and H. Schulzrinne. "vDelay: A tool to measure capture-to-display latency and frame rate," *Multimedia, International Symposium* on, 0:194-200, 2009.

[3] E. Ohbuchi, H. Hanaizumi, and L. A. Hock, "Barcode Readers using the Camera Device in Mobile Phones," *Proceedings of the 2004 International Conference on Cyberworlds*, pp. 260-265, November 2004.

[4] R. I. McCartney, "A liquid crystal display response time compensation feature integrated into an lcd panel timing controller," in *Proc. SID Dig.,* 2003, pp. 1350–1353.

[5] Jacobson, Ralph E. The manual of photography: photographic and digital imaging. Focal Press, 2000.

[6] M. Menozzi, F. Lang, U. Naepflin,C. Zeller, & H. Krueger, "CRT versus LCD: Effects of refresh rate, display technology and background luminance in visual performance," *Displays* 22.3 (2001): 79-85.

[7] Bss QR Code Generator SDK. http://www.barcodesoftwaresolutions.com.

[8] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Biomed. Eng.*, vol. BME-9, pp. 63–66, 1979.

[9] ZXing barcode reader. http://code.google.com/p/zxing/.