Camera-based Video Synchronization for a Federation of Mobile Projectors

Kiarash Amiri, Shih Hsien Yang, Christopher Larsen
Fadi Kurdahi, Magda El Zarki, Aditi Majumder
University of California, Irvine
{kamiri,shihhsy,cblarsen,kurdahi,elzarki,majumder}@uci.edu

Abstract

Ultra-portable projectors, called pico projectors, are now being embedded in mobile devices like cell-phones. Such mobile devices are projected to be the primary device to be used by younger people for ubiquitous sharing of all possible media initiating novel social interaction paradigms. Yet, the pico-projectors offer a much lower resolution and brightness than a standard projector. However, images displayed from multiple such mobile devices can be tiled to create a dramatically improved display in both brightness and resolution. This will allow multiple users to view and share media at a much higher quality.

In this paper, we present a camera-based video synchronization algorithm that allows a federation of projection-enabled mobile devices to collaboratively present a synchronized video stream, though only a smaller part of the video comes from each device. Since, the synchronization does not use any wireless network infrastructure, it is independent of network congestion and connectivity. We combined our method with existing distributed registration techniques to demonstrate a synchronized video stream for a federation of four projectors arranged in a 2×2 array. To the best of our knowledge, this is the first time that a camera-based technique has been used to mitigate network uncertainties to achieve accurate video synchronization across multiple devices.

1. Introduction

The first wave of ultra-portable projectors, called pico projectors (Figure 1), often less than an inch thick, is now beginning to appear in the market. It is a response to the emergence of compact portable devices such as mobile phones, personal digital assistants, and digital cameras, which have sufficient storage capacity to handle presentation materials but little real estate to accommodate larger display screens. Pico projectors allow projecting larger digital images onto most common viewing surfaces, like a wall or table, for extended periods of time.

Figure 1. Actual Pico Projectors from TI (left), embedded in a product like the 3M pocket projector (center), the Samsung i7410 Pico Projector phone (right).

Figure 2. Social Interactions around the Pico Projectors.

The anticipated acceptance and success of the embedded projector phone is very high. Pico projectors are projected to be the primary device to be used by younger people for ubiquitous sharing of all possible media (Figure 2) initiating novel social interaction paradigms. In fact, iSuppli predicts that shipments of embedded pico projectors will grow sixtyfold from 50,000 units in 2009 to more than 3 millions in 2013 [9].

Pico projectors are the result of a tremendous advancement in both LED/laser based illumination and DLP technology. Pico-projectors boast a 20,000 hour lamp life [15], which translates to over 18 years of life if the projector is used 3 hours per day. They consume about 200 times less power than a standard projector (1.5 Watts as opposed to 300 Watts) and are about 25-35 times lighter than a standard projector (4-5oz as opposed to 7-11lbs). This unprecedented improvement in power-efficiency, weight and longevity can provide an illusion that embedded pico projectors are the answer to all our dreams. On the contrary, these come at the cost of a severely reduced image quality. The pico projector has 12 lumens brightness, about 300 times lower than the 2600 lumens of brightness of a common commodity projector; and a QVGA resolution of 0.08 Megapixels, about 25 times lower than the 2 Megapixels

HDTV resolution of a common commodity projector.

This poses a serious limitation when coupled with the fact that the current years have seen an explosion in the video resolution and quality of capture devices, thanks to inexpensive high-resolution and high-dynamic range cameras. Thus, the current generation of users are used to a much higher quality media content than what offered by the pico projectors.

However, unlike any other alternate display technology, pico projectors have a distinctive advantage – the image displayed from multiple pico projectors can be overlaid on top of each other or tiled to create a dramatically improved display in both brightness and resolution. This "overlay" ability enables a federation of multiple pico projectors to offer the unique ability to create a higher quality display than possible from a single pico projector thus allowing multiple users to view and share media in a much more acceptable fashion than is possible with any alternate display technology. When coupled with a suitably transparent interface to form the necessary federation, a display consisting of federated pico projectors can foster novel collaborative interactions – like several co-workers or business colleagues gathered informally to discuss a presentation or a group of young on the go users in an ad hoc social gathering watching a higher quality YouTube video or a high-resolution live sports or news event. This paper focuses on this widely anticipated scenario of viewing of high quality video by aggregating the output of multiple such devices.

1.1. Main Contribution

In this paper, we consider a federation of *tiled* pico-projectors (embedded in mobile devices) together creating a high resolution video, though the image quality from each is much inferior. We assume that these mobile devices also have embedded cameras which can see the projected display. The viewing experience of video for such a federation is critically dependent on the synchronization of the frames across the multiple devices. We desire a video synchronization technique that does not depend on congestion, connectivity and delay variability in the mobile network. In this paper, we design a novel video synchronization method based on the visual feedback offered by the embedded cameras. We make this visual feedback channel as the primary channel of synchronization and use the additional channels of network, Bluetooth or WiFi for assistive purposes. In this way, we not only avoid burdening the network with more data due to synchronization requirements, but also achieve a much faster synchronization that is independent of network dynamics.

We first present a centralized algorithm that runs on a designated master projector, only which needs to have a feedback camera. Next we extend this method to present a distributed SPMD algorithm (Section 3) where identical method runs on each projector, but collectively achieves the video synchronization across the tiled federation of pico-projectors. This method is more scalable and assures convergence though runs asynchronously on a federation of such devices. Finally, we show that this method can be easily integrated with existing methods that align the images from multiple projector to create one single seamless image. We demonstrate this method on a real federation of 2×2 array of four pico-projectors. To the best of our knowledge, this is the first time camera-based methods are being explored to synchronize frames of video across a federation of projectors.

1.2. Related Work

There is a large body of literature on multi-projector displays, relevant to the context of the federation of pico-projectors. These have focused on two aspects: the geometric and color registration across the display and the architecture used to display information and interact with it. Most earlier works on registration focus on centralized registration where a single master should handle the multiple projectors [3, 16, 17, 18, 19, 20, 27, 13, 14, 23, 22, 28, 12, 24]. The user is expected to define the array configuration to this master who is then responsible to get feedback from the camera(s) to register the image across them. However, such centralized approaches are particularly unsuitable for an ad-hoc federation of mobile devices. Recently, distributed methods have been developed for auto-registration of a federation of projector-camera-PC ensembles [2, 26, 21] — identical in architecture to our federation of pico-projectors. We integrate an adaptation of the auto-registration method proposed in [21] to the video synchronization method proposed in this paper.

In parallel, we have seen the development of distributed rendering methods [4, 5, 6] where the rendering takes place in a distributed manner in computers attached to each projector, but they are controlled by a centralized server that manages how the rendering should be distributed. More recently, we have seen the development of distributed interaction paradigm [21] where a single program multiple data (SPMD) algorithm on each projector detects, tracks and reacts to a user action in completely distributed manner affecting only the projectors that see the gesture and are required to react. This assures minimal network bandwidth usage since all projectors do not communicate to a single centralized server and minimal time since the processing is shared by multiple projectors and is not the responsibility of a single centralized server.

However, all these works have not considered synchronization issues. All earlier works consider multiple projectors in a LAN setting where often the machines driving the projectors are usually dedicated to the display with not much CPU or network load. Further, such multi-projector

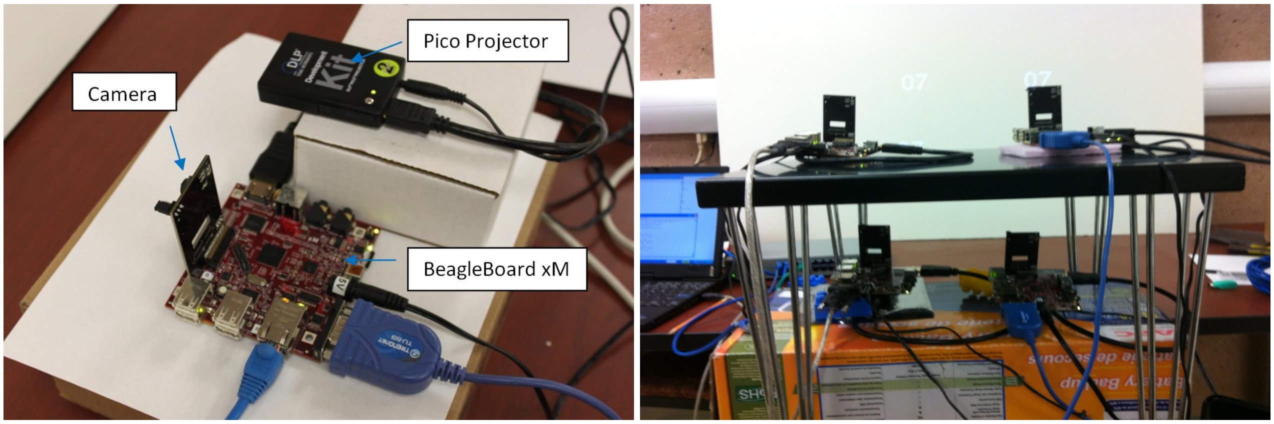

Figure 3. Left: One of our pico-projectors connected to the development board and equipped with a camera facing the projection area. Right: Setup of 4 tiled pico-projectors.

displays till date were mostly used for creating very large displays where the human field-of-view can never focus on the entire display at the same time. Hence, small latency across the different displays was below the threshold of human perception and went unnoticed. Hence, there is not much prior work addressing synchronization issues across multiple projectors. Loose synchronization has been achieved via NTP (network time protocol) in these works. Since NTP provides reasonable synchronization in a LAN setting, this has been sufficient for the current systems. When considering a federation of projector-embedded mobile devices on a heavily congested mobile network, synchronization becomes a practical issue which can completely ruin the viewing experience. This is especially true in this very small-sized displays (approximately 17-19” diagonal) since the user field-of-view can allow focusing on all the displays at the same time. Further, videos demand a 30 frames per second synchronization, a very stringent requirement given the congestion in mobile networks. This motivated us to look for alternate modalities for achieving video synchronization. This paper makes the first effort to explore the option of using the local visual feedback from the already existing camera on the mobile device for synchronization purposes.

2. System Overview

Our setup consists of multiple tiled pico projectors each connected to a development board and equipped with a camera facing the projection area (Figure 3). Projectors are tiled together overlapping at the boundaries. Each projector thus shows only a spatially segmented video. Our projector and camera on the mobile device need not be gen-locked with each other. We assume that the camera capture rate is more than double the projector display frame rate i.e.super-Nyquist sampling. This assures that we can capture images by the camera that do not span across two projector frames resulting in ambiguity.

We introduce two visual synchronization schemes to synchronize the display time of different partitions of a frame projected by different projectors: (a) a centralized synchronization; and (b) a distributed synchronization.

In the centralized setup one processing unit acts as the master and runs the synchronization algorithm. In this scheme only the master needs to have a camera which should cover the whole projection area however a communication channel between the master and other boards is needed to transfer the calculated synchronization parameters to each corresponding board. It is important to note that this communication occurs after synchronization calculations and any congestion or delay related to communication between master and other boards does not affect the synchronization accuracy. This method can be used even if all the mobile devices are not equipped with a front facing camera.

In a distributed scheme each unit needs to be equipped with a camera and it independently runs the synchronization algorithm through the visual feedback from its camera. We assume that the camera on each projector sees the entire display. This is a reasonable assumption for these small format displays. Even 4 pico projectors together creates a 19” diagonal display which easily comes within the field-of view of the camera. In this scheme each projecting unit autonomously adjusts itself to achieve synchronization, and there is no need for a communication network between the boards. In the following section, we present these two solutions.

3. Algorithm

In a single projector environment, as the device starts the video playback by displaying the first frame, accurate display time of the subsequent frames can be calculated from its internal clock rate such that the target frame rate for the playback can be achieved. In a setup of multiple tiled projectors, we have to assure the following: first, as in the case of a single projector, internal to each projector the periodic display of frames occurs at the correct time realizing a target frame rate. Secondly, it is also necessary to match the display time of the same frame across the projectors to implement a synchronized playback on the tiled display. Once this synchronization is achieved, by displaying frames at the correct display time based on the internal clock rate of the individual projector, the synchronization can be maintained across the entire video sequence. This assumes that the clock drift in each device is negligible during the playback which is an acceptable assumption considering a 0.5 ppm stability for oscillators [11] available in the market for mobile devices that results in less than 2ms drift in an 1 hour playback period. Hence, the synchronization can be achieved as a preprocessing before the actual start of playback on the setup of multiple projectors. We also assume that the synchronization is preceded by a registration procedure [26, 2, 21] which recovers the ID for each projector. In a system with n projectors, the projector ID is an integer between 1 to n.

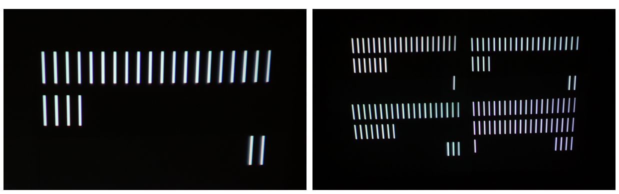

Figure 4. Left: Coded patterns projected from each projector during synchronization period. Right: The pattern captured by the master camera.

3.1. Centralized Synchronization

In this synchronization scheme only one mobile device needs to be equipped with camera. This device acts as a master and runs the centralized algorithm calculating the delays needed to synchronize all the projecting units. The camera on the master device should cover the whole projection area. Initiated by the master, the synchronization process begins by having each projector start projecting a sequence of frames at a target frame rate where each frame is an otherwise blank frame with the frame number and the projector ID encoded as a pattern (e.g. every 33ms for 30fps) (Figure 4). We refer to this sequence of frames as the synchronization sequence. After projection has started on all the projectors, the camera corresponding to the master unit captures an image that contains the frames projected by all projectors at an arbitrary time. Figure 5 shows an example of 4 out of sync projectors that started displaying the synchronization sequence at different times and the red line shows the master camera capturing an image. The captured image is then processed to find the projector with the minimum frame number (maximum frame lag). This projector is used as the synchronization reference. For each of the other projecting units, the master computes the reference projector frame lag L from the unit's projector and informs the unit of this lag over the network. Each projector stalls its current frame for the next L frames, as shown in Figure 5. Thus, the maximum time difference between any two projectors displaying the same frame can be brought down to less than a frame period.

Algorithm 1 *Pseudo code for the Master unit and Projecting devices in centralized synchronization.*

Master

Send the start synchronization command to all projecting devices with registered IDs
Wait until all devices respond that they have started projecting synchronization sequence
Capture an image from the projection area (covering all projectors)
Decode the coded patterns in the image and extract the device IDs and corresponding frame numbers
Find the ID for the most lagging device which has minimum frame number in the captured image
For each device find the required stalls as the difference between its captured frame number and the lagging device frame number
Based on their IDs send the stalls to each device

Other projecting devices

Start initiated by the master
Read the internal time
Display the first synchronization frame that is a coded pattern containing device ID and frame number 1
Notify master about starting to display the synchronization sequence
while not end of video playback **do**
    Wait for next display time based on reading internal time
    if finished synchronization sequence **then**
        show next decoded video frame
    else
        if received stalls from the master and stalls needed is greater than zero **then**
            Repeat displaying previous coded pattern
            Decrement stalls needed
        else
            Show coded pattern for the next frame number
        end if
    end if
end while

Note that the time taken for the communication does not affect the quality of synchronization, as shown in Figure 6. It merely affects the number of frames required to achieve synchronization. The pseudo code for the master and other projecting units is given in Algorithm 1.

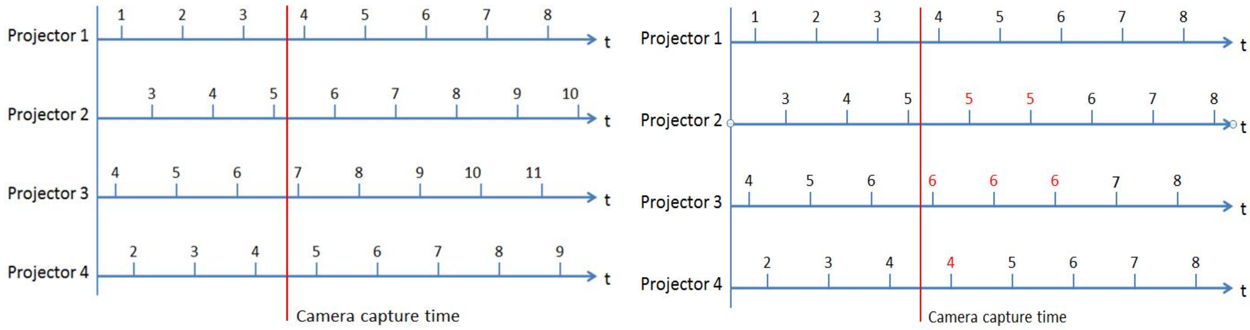

Figure 5. Left: Frames being displayed at a target frame rate by 4 out of sync projectors. At a given time that master captures an image projector 1 is displaying frame 3, projector 2 is displaying frame 5, projector 3 is displaying frame 6 and projector 4 is displaying frame 4. The most lagging device in this case is the first projector. The red line shows the master camera capture time. The computed lags in projector 2, 3 and 4 are 2, 3 and 1 respectively. Right: After the lag is communicated, projector 2, 3 and 4 stall for 2, 3 and 1 frames respectively. Thus, when displaying frame 7, all the projectors are synchronized.

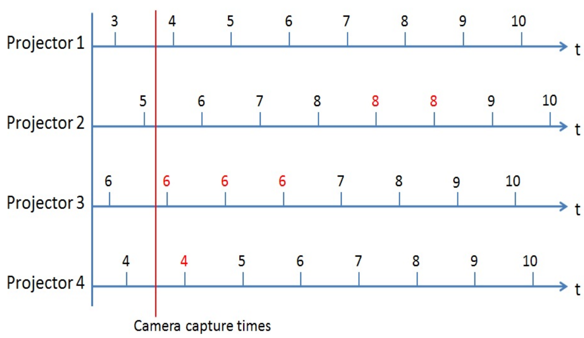

Figure 6. This shows the effect of delay in communication in synchronization of Figure 5. If the message to projector 2 reaches 3 frame later due to network congestion, synchronization is achieved in frame 9 instead of frame 7.

While devices in our setup may have different physical clocks with different oscillator rates, since we are using the calculated time of each device to determine the frame display times, our approach works even though the clock rates across the projectors are different.

3.2. Distributed Synchronization

The centralized synchronization uses a single master device, however it needs to be able to communicate the calculated stalls to each projector device. In a case where each projecting unit has its own camera to capture the whole projection area, the synchronization task can be distributed between devices and the communication requirement between the units is eliminated.

In the distributed approach all devices run the same algorithm and adjust themselves individually to achieve synchronized state. In this scheme each device does its own image capture. It identifies itself using the embedded device IDs and also identifies the device with the most lag in time (with smallest frame number) using the embedded frame numbers. Then using the captured frame number difference between itself and the lagging device, it calculates the lag L of the *most* lagging device from itself. It then stalls for the next L frames internally during its frame buffer handling process to let the device with the highest lag in time catch up.

Algorithm 2 *Pseudo code for Projecting devices in distributed synchronization.*

```
Initialize stalls to zero
Read internal time
Display the first synchronization frame which is a coded pattern containing device ID and frame number 1
while not end of video playback do
    Wait for next display time based on reading internal time
    if finished synchronization sequence then
        show next decoded video frame
    else
        if stalls needed is greater than zero then
            Repeat displaying previous coded pattern
            Decrement stalls needed
        else
            Show coded pattern for the next frame number
            if did not capture an image by camera before then
                Capture an image and decode coded patterns
                Find the most lagging device with smallest frame number
                Update the stalls variable with the difference between the frame number of yourself and the lagging device
            end if
        end if
    end if
end while
```

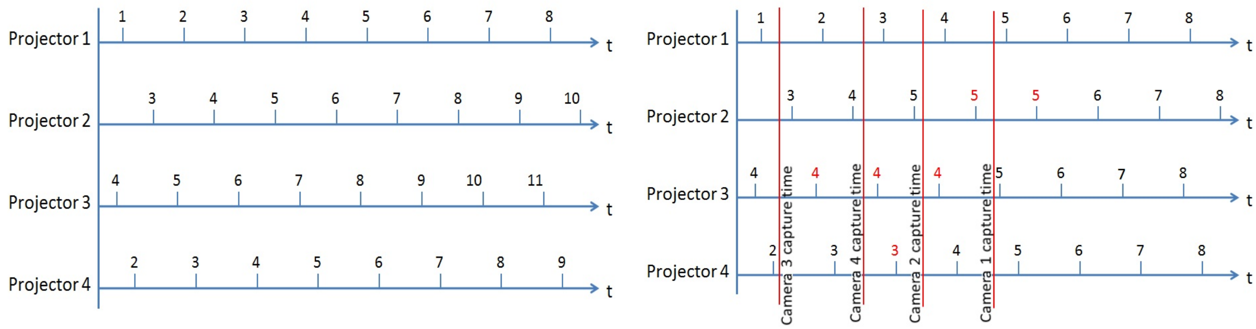

Figure 7. Left: Frames being displayed at a target frame rate by 4 out of sync projectors. Right: Red lines show the capture time of cameras covering all 4 projectors with the same relative timing as shown in the left side image. After calculating the lags and applying the corresponding stalls in each device, all the projectors are synchronized starting at displaying frame 6.

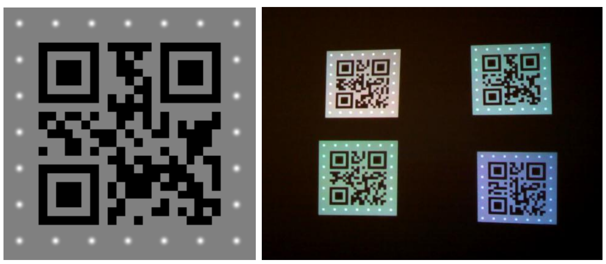
Figure 8. Left: QR codes embedded frame projected during synchronization and calibration of 4 tiled pico-projectors. Right: Image captured by one of the cameras during the synchronization and calibration process.

Thus, after a pre-specified number of synchronization frames, all projectors start the regular video playback while projecting synchronized with each other, as shown in Figure 7). Pseudo code for the SPMD (single process multiple data) distributed synchronization scheme is illustrated in Algorithm 2.

3.3. Integration with Registration Techniques

[21] presents an algorithm to achieve distributed registration across multiple projectors. This method uses QR codes augmented with some gaussian blobs as patterns (Figure 8). These codes encode certain information. The cameras capture these codes and decode them to find the configuration of the display (total number of projectors, their configuration in number of rows and columns, and the projector's own coordinates in the array). The embedded blobs are used to find homography across adjacent projectors and a radially cascading method is used to register the images across the multiple projector geometrically. The homography is also used to achieve an edge blending across the overlaps. This registration is also achieved once before video playback starts.

Since both the temporal synchronization and registration are designed to occur before the actual playback we can combine the two. Fortunately, the QR codes used in [21] still has empty channels which can be used to embed our frame number information for synchronization. So, we augment these same QR codes used for registration to achieve our synchronization. Thus, we integrate the registration of [21] and our synchronization to happen as a single process before the video playback starts.

3.4. Handling Sub-Nyquist Camera Capture Time

In most practical systems, the camera and projector frame rate are comparable. Hence, most of the time the camera sampling rate is sub-Nyquist when compared to display rate. Hence, there is a high chance that the capture duration from the camera spans multiple projector frame. Since seldom the camera capture rate is less than half of the display frame rate, the camera capture duration, more often than not, spans two frames. During synchronization, this implies that the camera captures two different QR codes with two different frame numbers within the same capture time. Hence, deciphering the QR code to decode the frame number becomes difficult. To alleviate this, we place the QR codes in two non-overlapping spatial region in alternate frames. Thus, even if the captured image spans multiple projector images, the captured image has two spatially separated QR code (Figure 9. Both these codes are decoded to extract the frame number and only the high frame number is retained for lag computations. Note that this does not affect the registration since the information pertinent to registration remains identical across both the QR codes captured by the camera. Also, the number of blobs captured doubles which increases the number of correspondences used for registration and can only result in a better registration accuracy.

4. Implementation and Results

We implemented our algorithm for a setup of four tiled projectors as shown in Figure 10. We used Texas Instruments DLP Pico Projector 2.0 Development Kit [8], BeagleBoard-xM development board [1] and 3 MegaPixel Leopard Imaging Camera board [7] designed for

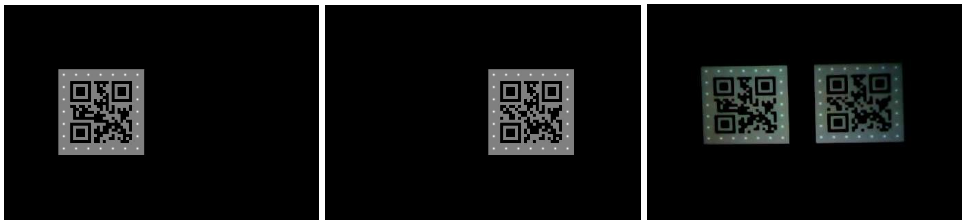

Figure 9. QR codes embedded in an odd numbered (left) and even numbered (middle) projected frame to handle sub-Nyquist camera capture. Image captured (right), when spanning across two projected frame, by one of the cameras during the synchronization and calibration process.

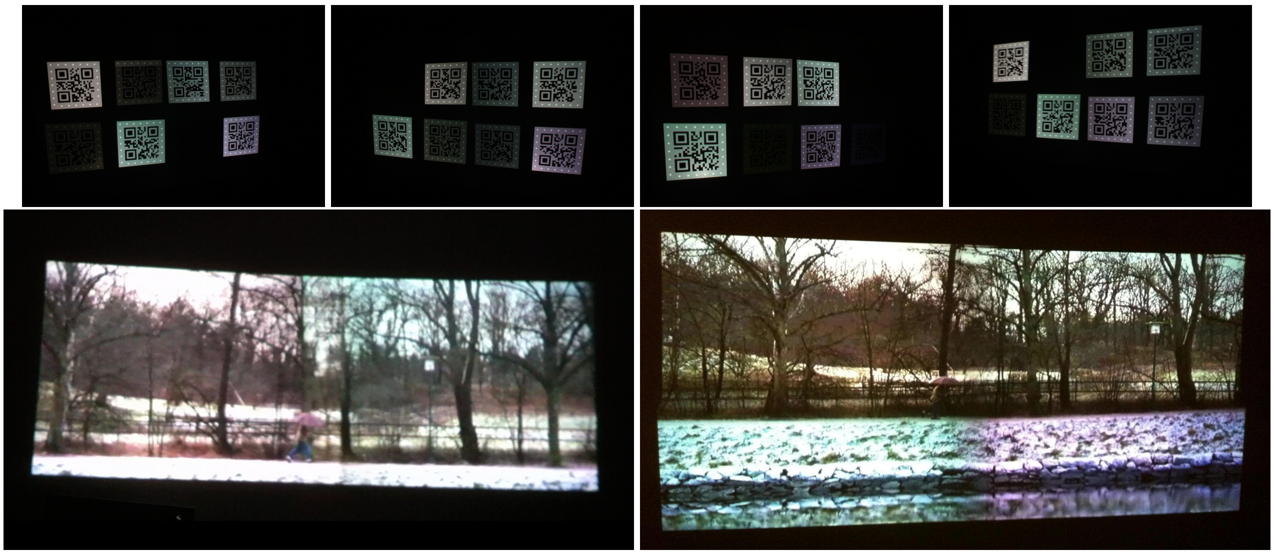

Figure 10. Top: The captured frames from the four cameras on a four projector setup which is used to achieve synchronization. Bottom: Video after synchronization and registration on a 2 projector (left) and a 4 projector (right) system.

BeagleBoard-xM platform to demonstrate a prototype (figure 3) for mobile devices equipped with camera and pico-projector.

We have achieved synchronization of less that 33ms using our method. Though this does not achieve clock synchronization at a much final granularity of micro or nano seconds, this is sufficient for our purpose. The human visual persistence is around one tenth of a second. Earlier works on psychophysical analysis [25, 10] related to the response time and display rate in human performance with computers use this fact to show that in most situations users expect and can detect responses within a tenth of a second, i.e. 100ms (duration worth 3 frames considering a interactive rate of 30 fps). This is one of the primary reason that when using large scale multi-projector displays, a couple of frames latency have not been a big concern. However, it is true that being of much smaller formats, the displays from the mobile devices may push this tolerance down. Our synchronization of 33ms is hence already much lower than the sufficient threshold of 100ms. Further, in practice, we do not perceive any lag from this granularity of synchronization.

5. Conclusion

In conclusion, we proposed to use visual feedback from camera to synchronize video partitions displayed by multiple pico projectors. While we discussed our algorithm in the context of mobile pico projectors the same techniques are applicable in synchronization of standard projectors or even tiled displays. We introduced both the centralized and distributed synchronization schemes based on which we were able to synchronize video frame with accuracy of one frame period.

Our proposed method is just the first work in this direction and has a tremendous potential to be used in several directions. First, with the anticipated popularity of the mobile devices with embedded pico-projectors, it is easy to envision more than four projector systems (maybe eight or ten) tiled together to view a video. Or, the size of the projected imagery can also increase with more technological advancement in the design of such mobile projectors. In such scenarios, the camera on each mobile device may not be able to see the entire display, as is assumed in this paper. We are currently exploring extension of our method to handle such scenarios. Second, users may choose to superimpose such projectors (instead of tiling) to alleviate their low brightness. This will bring forth stricter synchronization requirements, especially for video. Further, the codes from multiple projectors will superimpose when decoding them will be difficult. We are also exploring adaptations of our method for such situations. Finally, synchronizing audio from multiple devices along with the video is also a challenge which we would like to explore.

References

[1] beagleboard.org. Beagleboard-xm. http://www.beagleboard.org.

[2] E. Bhasker, P. Sinha, and A. Majumder. Asynchronous distributed calibration for scalable reconfigurable multi-projector displays. *IEEE Transactions on Visualization and Computer Graphics (Visualization)*, 2006.

[3] H. Chen, R. Sukthankar, G. Wallace, and K. Li. Scalable alignment of large-format multi-projector displays using camera homography trees. *Proc. of IEEE Vis*, 2002.

[4] G. Humphreys, I. Buck, M. Eldridge, and P. Hanrahan. Distributed rendering for scalable displays. *Proceedings of IEEE Supercomputing*, 2000.

[5] G. Humphreys, M. Eldridge, I. Buck, G. Stoll, M. Everett, and P. Hanrahan. Wiregl: A scalable graphics system for clusters. *Proceedings of ACM SIGGRAPH*, 2001.

[6] G. Humphreys, M. Houston, R. Ng, R. Frank, S. Ahem, P. Kirchner, and J. Klosowski. Chromium : A stream processing framework for interactive rendering on clusters. *ACM Transactions on Graphics (SIGGRAPH)*, 2002.

[7] L. I. Inc. Leopardboard 365 3m camera board. http://www.leopardimaging.com.

[8] T. Instruments. Dlp pico projector development kit. http://www.focus.ti.com/docs/gencontent.tsp?contentId=52770.

[9] iSuppli Market Intelligence. Huge growth set for pico projectors in cell phones, other mobile electronics. http://www.isuppli.com.

[10] P. A. Laplante. *Real-Time Systems Design and Analysis*. IEEE Press 2nd edition, 1997.

[11] R. Limited. Rakon tcxo products for mobile applications. http://www.rakon.com/Products/Pages/TCXOs.aspx.

[12] A. Majumder, Z. He, H. Towles, and G. Welch. Achieving color uniformity across multi-projector displays. *Proceedings of IEEE Vis*, 2000.

[13] A. Majumder and R. Stevens. Color nonuniformity in projection-based displays: Analysis and solutions. *IEEE Transactions on Vis and Computer Graphics*, 10(2), March–April 2003.

[14] A. Majumder and R. Stevens. Perceptual photometric seamlessness in tiled projection-based displays. *ACM TOG*, 2005.

[15] Optoma. Optoma pico projectors. http://www.optomausa.com/productcategory.asp.

[16] A. Raij, G. Gill, A. Majumder, H. Towles, and H. Fuchs. Pixelflex 2: A comprehensive automatic casually aligned multi-projector display. *IEEE PROCAMS*, 2003.

[17] A. Raij and M. Polleyfeys. Auto-calibration of multi-projector display walls. *Proc. of ICPR*, 2004.

[18] R. Raskar. Immersive planar displays using roughly aligned projectors. In *Proc. of IEEE VR*, 2000.

[19] R. Raskar, J. V. Baar, T. Willwacher, and S. Rao. Quadric transfer function for immersive curved screen displays. *Eurographics*, 2004.

[20] R. Raskar, M. Brown, R. Yang, W. Chen, H. Towles, B. Seales, and H. Fuchs. Multi projector displays using camera based registration. *Proc. of IEEE Vis*, 1999.

[21] P. Roman, M. Lazarov, and A. Majumder. A saclable distributed paradigm for multi-user interaction with tiled rear projection display walls. *IEEE Transactions on Visualization and Computer Graphics*, 2010.

[22] B. Sajadi, M. Lazarov, A. Majumder, and M. Gopi. Color seamlessness in multi-projector displays using constrained gamut morphing. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2009.

[23] B. Sajadi and A. Majumder. Markerless view-independent registration of multiple distorted projectors on vertically extruded surface using a single uncalibrated camera. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2009.

[24] B. Sajadi and A. Majumder. Scalable multi-view registration for multi-projector displays on vertically extruded surfaces. *Proceedings of EuroVis*, 2010.

[25] B. Shneiderman. Response time and display rate in human performance with computers. *Computing Surveys*, 16(3):265–285, 1984.

[26] P. Sinha, E. Bhasker, and A. Majumder. Mobile displays via distributed networked projector camera pairs. *Projector Camera Systems Workshop*, 2006.

[27] R. Yang, D. Gotz, J. Hensley, H. Towles, and M. S. Brown. Pixelflex: A reconfigurable multi-projector display system. *Proc. of IEEE Vis*, 2001.

[28] R. Yang, A. Majumder, and M. Brown. Camera based calibration techniques for seamless multi-projector displays. *IEEE TVCG*, 2005.