Collaborative Video Playback on a Federation of Tiled Mobile Projectors enabled by Visual Feedback

Kiarash Amiri, Shih-Hsien Yang, Fadi Kurdahi, Magda El Zarki, Aditi Majumder

University of California, Irvine

{kamiri,shihhsy,kurdahi,elzarki,majumder}@uci.edu

ABSTRACT

Pico projectors are expected to become increasingly popular in the near future, in particular when embedded in mobile devices such as smart phones, portable media players and digital cameras. Consumers seeking more features and attracted to all in one portable devices, will find such integrated devices very appealing. However the resolution and the brightness provided by integrated mobile projectors are much lower than what standard projectors commonly offer. Our collaborative scheme based on our synchronization technique provides a means of increasing resolution and brightness for the video projected by such mobile devices, significantly enhancing the viewing experience for the user.

In this paper, we present a collaborative video playback on mobile projectors, set out and managed only through visual feedback. More specifically we introduce a camera-based video synchronization algorithm that allows a federation of projection-enabled mobile devices to collaboratively present a full video stream that consists of multiple sub streams, each streamed to a different mobile device constituting the ensemble. Since the synchronization does not use any wireless network infrastructure, it is independent of network congestion and connectivity. We combined our synchronization method with existing distributed registration techniques to demonstrate a synchronized video playback for a collaborative federation of four projectors arranged in a 2×2 array. To the best of our knowledge, this is the first time that camera-based techniques have been used to mitigate network uncertainties to achieve accurate video synchronization across multiple devices.

Keywords

Pico Projector, Video Synchronization, Mobile Video, Tiled Projectors, Mobile Projector

1. INTRODUCTION

The first wave of ultra-portable projectors, called pico projectors, often less than an inch thick, is now beginning to appear in the market. It is a response to the emergence of advanced portable devices like smart phones which have sufficient processing power and storage capacity to handle large 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 an extended period of time.

Figure 1: . ***Samsung Galaxy Beam i8520 Pico Projector phone (left), Samsung i7410 Pico Projector phone (center), and Nikon COOLPIX Camera (right).***

iSuppli predicts that shipments of embedded pico projectors will grow sixtyfold from 50,000 units in 2009 to more than 3 million in 2013 [8]. Figure 1 shows two currently available models of Samsung mobile phones which are already equipped with integrated projectors. Nikon COOLPIX S1000pj series of digital cameras is another example of devices with embedded projectors that are already in the market.

The low power consumption of LED/lasers allows pico projectors to operate within a severely rationed power supply - inevitable in portable devices. Advanced DLP technology removes the need for a color filter allowing extreme miniaturization and reduction in weight so that they can be embedded within small devices like cell phones or media players. This unprecedented reduction in power-consumption, size and weight can produce an illusion that embedded pico projectors are the answer to all our dreams.

This is far from the truth. The portability and power efficiency of the pico projectors are traded with their brightness and resolution therefore resulting in a severely reduced image quality. Compared to 2600 lumens for a common commodity projector, the pico projector emits around 10 lumens of light, providing about 300 times lower brightness for similar size image. Compared to a resolution of 2 Megapixels (at HDTV resolution of 1920×1080) of a common commodity projector, an embedded pico projector offers a small resolution of 0.15 Megapixels (at HVGA resolution of 480×320) [2] or about 15 times lower. 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. The current generation of users is accustomed to a much higher quality media content than what the pico projectors are offering today or will be able to offer in the foreseeable future.

However, unlike any other alternate mobile 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 possibility 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. 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 mobile 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 all projectors. 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 and we make this visual feedback channel as the primary channel of synchronization. In this way, we not only avoid burdening the network with more data due to synchronization requirements, but also achieve an accurate synchronization that is independent of network dynamics. It is also worth mentioning that as embedded cameras get cheaper and smaller and in response to new application requirements, we are seeing more cameras getting integrated in the mobile devices. For example Apple iPhone first started with one integrated camera, then for the Facetime application a second camera was integrated in the new generation of iPhones. Now in the new Apple patent application [6] possibly for the next generations of iPhones equipped with integrated projectors, another camera which is facing the projection area of projector has been considered for interaction based applications which benefit from the visual feedback of that camera (e.g. gesture based interactions for data sharing and transfer). We use such camera to provide feedback for our scheme.

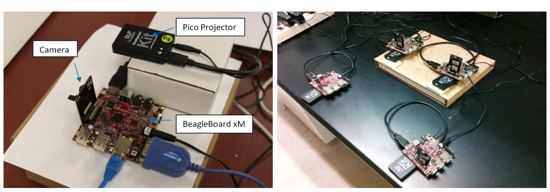

Figure 2: ***Left: Setup of Pico projector connected to the development board equipped with a camera. Right: Setup of 4 tiled pico projectors.***

We initially introduce the concept behind our visual feedback based synchronization using a centralized algorithm that runs on a designated master projector, only which needs to have a feedback camera. Next we present a technique to achieve higher synchronization accuracy while using the proposed scheme. Then we extend our synchronization method to present a distributed SPMD (Single Program Multiple Data) algorithm where identical method runs on each projector, but collectively achieves the video synchronization across the tiled federation of pico projectors. We show that this method can be easily integrated with existing methods that align the images from multiple projectors to create one single seamless image. Finally we extend our distributed design to present a scalable scheme that assures convergence though runs asynchronously on the federation of mobile projectors. We demonstrate our method on a real federation of 2×2 array of four pico projectors (Figure 2). 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 handles the multiple projectors [4, 10, 11, 14, 9]. The user is expected to define the array configuration to the master who is then responsible to get feedback from 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 [3, 13, 12] — identical in architecture to our federation of pico projectors. We integrate an adaptation of the auto-registration method proposed in [12] to the video synchronization method proposed in this paper.

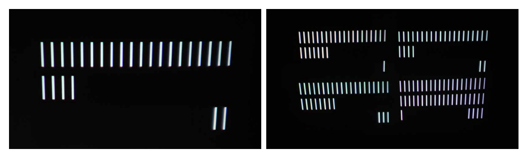

Figure 3: ***Left: Bar patterns projected by each projector during synchronization period. Right: Image captured by the master camera.***

However, current body of work has not considered synchronization issues especially among mobile devices. All earlier works consider multiple projectors in a wired LAN setting where the machines driving the projectors are usually dedicated to displaying the image with little to no other CPU or network load. Synchronization was achieved via NTP (network time protocol) or networked barriers in some of these works. Since such techniques provide reasonable synchronization in reliable wired LAN settings, that was considered satisfactory performance for earlier systems [7, 5]. When considering a federation of projector-embedded mobile devices on a heavily congested mobile network, synchronization becomes a practical issue. In this paper our scheme proposes using local visual feedback from the mobile device's camera for synchronization purposes.

2. ALGORITHM

In a single projector device environment, as the device starts the video playback by displaying the first frame, accurate display times for the subsequent frames can be calculated from its internal clock such that the target frame rate for the playback can be achieved. In a setup of multiple tiled projectors however, we have to assure both the following: first, as in the case of a single projector, internal to each projector the periodic display of frames occurs at correct times realizing the target frame rate. Secondly, it is also necessary to match the display times of the partitions 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 times based on the internal clock of the individual projector device, the synchronization can be maintained across the entire video sequence.

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

```
Master
Send the start synchronization command to all projectors
Wait until all devices respond that they have started projecting
  synchronization sequence
Capture an image from the projection area
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 be-
  tween 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 se-
  quence
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
```

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 [1] available in the market for mobile devices that results in less than 2ms drift in an 1 hour playback period. Using our proposed scheme, the synchronization can be achieved before the actual start of playback on the ensemble of multiple projectors. We also assume that the synchronization is preceded by a registration procedure [13, 3, 12] which recovers the ID for each projector. In a system with n projectors, the projector ID is an integer between 1 to n.

2.1 Camera Feedback based Synchronization

In the most simple implementation of our scheme, only one mobile device needs to be equipped with a camera. This device acts as a master and runs the centralized algorithm calculating the delays needed to synchronize all the projecting units. For this scenario, the field of vision of the camera on the master device must 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 (e.g. every 33 ms for 30 fps) where each frame is an otherwise blank frame with the frame number and the projector ID encoded as a pattern (Figure 3). 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 the projectors at an arbitrary time. Figure 4 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 identify the projector displaying the minimum frame number (which has the 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 4. Thus, the maximum time difference between any two projectors displaying the same frame can be brought down to less than a *frame display period*.

Note that the time taken for the communication does not affect the quality of synchronization, 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.

2.2 Sub-Frame Synchronization Accuracy

The accuracy of the proposed synchronization technique can be further enhanced to below a video frame period. That can be achieved if the camera used for synchronization has a higher rate than the Nyquist rate of the video playback frame rate. In that case, as long as the Nyquist rate and the projector frame rate limits allow us, we can run projectors at higher display rate during the *synchronization period* then at the start of the video playback we switch to the actual target video frame rate to achieve further accurate synchronization. Alternatively, as we describe below, to realize a sub frame granularity of synchronization, the display rate of the projectors can stay constant at the target video frame rate while more images per frame are captured during the *synchronization period*. Consider m projector devices projecting frames all at the same frame rate f frames per second and assume that a camera with capture rate of $n \times f$ frames per second captures n images during a projector frame period $(1/f)$. The same way as described before, the first captured image is used to synchronize any given projector with the lagging projector with the synchronization accuracy of one frame period. The number of stalls needed for each projector can be calculated at the central unit as $s_i = fr_i^1 - fr_{min}^1$ where s_i is the number of stalls needed for projector i; fr_i^1 is the frame number corresponding to projector i extracted from the first captured image; and fr_{min}^1 is the minimum frame number corresponding to the reference projector (lagging projector) extracted from the first captured image. The frame display times are now within a frame period. After this preliminary synchronization, the frame display time of the projectors can be further adjusted by a fraction of the frame period to increase the synchronization accuracy. Once the projectors are within a frame lag, we require one more round to detect the projector that is furthest behind within that one frame period. As before, all projectors delay their display times to match the one with the highest lag in time however this time within the given single frame period.

Since all n images captured by the camera are taken during one frame period of projection, at most 2 consecutive synchronization frame numbers from each projector will appear in the captured images. In other words, for a given projector i:

$fr_i^1 = fr_i^2 = ... = fr_i^l =$ value of the first captured frame number of projector i

$fr_i^{(l+1)} = fr_i^{(l+2)} = ... = fr_i^n =$ value of the second captured frame number of projector i

In the above statement, l, where $1 < l < n$, is the number of times

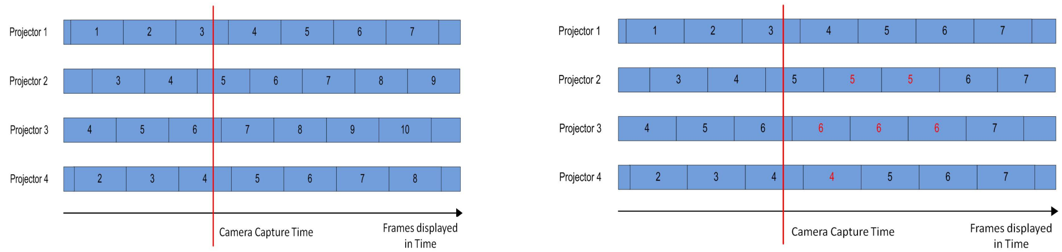

Figure 4: ***Left: Frames displayed by 4 out of sync projectors. At a given time that master captures an image, projectors 1,2,3 and 4 are displaying frames 3,5,6 and 4 respectively. The device with the highest lag in this case is the first projector. The computed lags for projector 2, 3 and 4 are 2, 3 and 1 respectively. Right: After the lag is communicated, projectors stall accordingly. Thus, when displaying frame 7, all the projectors are synchronized.***

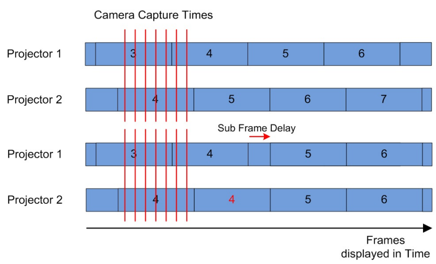

Figure 5: ***Top: Camera capture times and display time of frames without applying delays. Bottom: Projector 1 is the lagging projector at frame level therefore projector 2 is stalled 1 frame. At sub frame level*** $n = 7$***,*** $l_{max} = 7$ ***(corresponding to projector 2) and*** $l_1 = 5$ ***therefore*** $d_1 = 2$ ***which means*** $2/(7 \times f)$ ***sub frame delay applied to projector 1 to achieve sub frame synchronization.***

the smaller frame number was captured from projector i and $n - l$ is the number of times the larger frame number was captured from projector i. Among projectors the one which corresponding smaller captured frame number has the most frequent appearance comparing to other projectors (projector with maximum l, l_{max}) will be the lagging projector after the frame level synchronization. By identifying the reference projector (lagging projector with l_{max}) sub frame delay d_i for a given projector i can be determined as: $d_i = l_{max} - l_i$. Figure 5 illustrates an example for two projectors.

Based on this, the more the number of captured images n during a frame period $1/f$, the better the accuracy of calculated delay for each projector would be, which results in lower sub frame synchronization error calculated as $1/(n \times f)$.

2.3 Network Independent Synchronization

The synchronization algorithm we introduced so far runs on a single master device, however it has to communicate the calculated frame stalls and sub frame delays to each projector device. In a case where each projecting unit has its own camera to capture the projection area, the synchronization task can be distributed between devices and the communication requirement between the units is eliminated.

In such distributed approach all devices run the same algorithm and adjust themselves individually to achieve the 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 locally for the next L frames to let the device with the highest lag in time catch up.

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

Initialize *stalls* to 0
Read the 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 lagging device with smallest frame number
        Set *stalls* variable as the difference between the frame number of yourself and the lagging device
      end if
    end if
  end if
end while

Thus, after a pre-specified number of synchronization frames, all projectors start the regular video playback synchronized with each other, as shown in Figure 6. Pseudo code for the SPMD distributed synchronization scheme is illustrated in Algorithm 2.

2.4 Integration with Registration Techniques

In [12], the authors present an algorithm to achieve distributed registration across multiple projectors. This method uses QR codes augmented with gaussian blobs as patterns (Figure 7). These codes contain required information regarding each projector. The cameras capture these codes and decode them to find the configuration of the display. 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 projectors geometrically. The homography is also used to achieve an edge blending across the overlaps. This registration is also applied before the video playback starts.

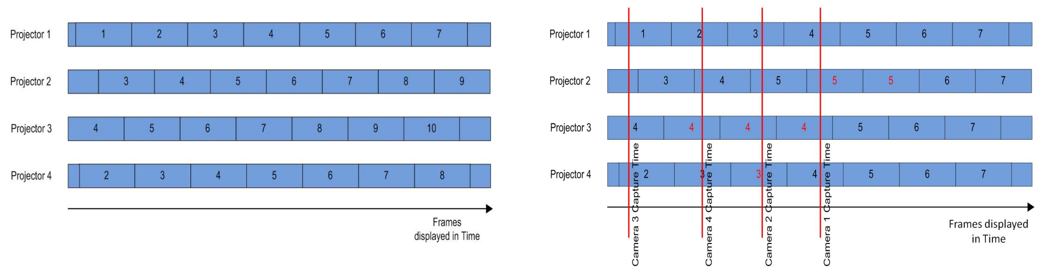

Figure 6: ***Left: Frames displayed by 4 out of sync projectors. Right: Red lines show the capture time of cameras covering all 4 projectors. After calculating the lags and applying corresponding stalls in each device, all the projectors are synchronized starting at displaying frame 6.***

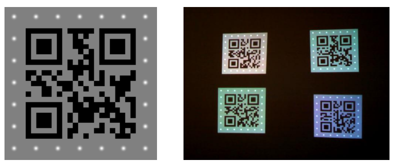

Figure 7: ***Left: QR code projected by each device during synchronization and registration of 4 tiled pico projectors. Right: Image captured by one of the device cameras.***

Since both the temporal synchronization and the registration are designed to occur before the actual playback we can combine the two. Frame numbers displayed during synchronization period along with other information needed for synchronization can be embedded in the same QR codes (instead of bar patterns) since these codes have enough capacity to store data needed for both operations. Thus, we integrate the registration [12, 13, 3] and the synchronization to occur as a single process before the video playback starts. Therefore the synchronization sequence is implemented as a sequence of QR codes containing information needed to perform both registration and synchronization.

2.5 Distributed Scalable Synchronization

The distributed synchronization as described earlier eliminates the communication requirement between a central unit and the projecting devices, however it still requires that each camera captures images containing projection area of all projector devices which may not be feasible as we increase the number of devices. Therefore in this section we present a scalable approach for the camera based synchronization of projectors in a distributed setting. Here again we assume all projectors are positioned in a tiled setup, however each camera accompanying a projector captures images containing its own projection area and the projection area corresponding to 3 other projectors: above it, to the left of it and the above left, instead of the whole projection area.

We use a flag for each projector to specify if the projector has joined the set of synchronized projectors and therefore can be used as a reference for synchronizing the rest of the projectors. This flag is embedded in the coded pattern (QR code) projected by the device and can be observed by the cameras of the projecting units to the right, to the bottom and to the bottom right of the device. Therefore those projector devices that see a synced reference projector can each synchronize themselves with the reference and update their flag as they become a reference for next projectors (Figure 8). The first reference is the top left projector. Pseudo code for scalable distributed synchronization has been given in Algorithm 3.

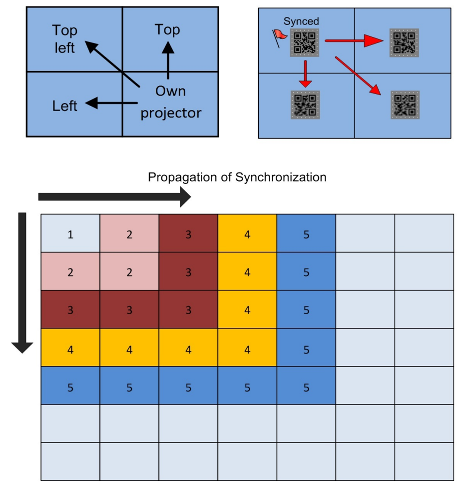

Figure 8: ***Top Left: Camera coverage of each device in scalable scheme. Top Right: Sync up flag seen by the neighbor devices. Bottom: Flow of synchronization. Each set of projectors can find a reference from previous set to sync up with.***

3. IMPLEMENTATION

To present the visual feedback based collaborative video playback on mobile projectors, we used a setup of four devices as we can expect a federation of four tiled projectors would be a common use case for collaborating mobile projectors. For that we implemented the distributed network independent synchronization algorithm. We used Texas Instruments DLP Pico Projector 2.0 Development Kits, BeagleBoard-xM development boards and 3 MegaPixel Leopard Imaging Camera boards to show prototypes (Figure 2) of mobile devices equipped with camera and pico projector.

Figure 9 shows images of the synchronization and the geometrical registration being applied to a four projector system. Based on the calculated homographies across projectors and the blending requirements on the overlapping areas, each unit constructs the video frame segment that it needs to project to collaboratively show the high resolution video frame (1200×720) as all projectors display their segments synchronous to each other with synchronization accuracy of 33ms.

4. CONCLUSIONS

In this paper we showed that we can use visual feedback from embedded cameras to synchronize video partitions displayed by

Figure 9: ***Left: QR code patterns displayed by projectors are used by a visual feedback based program to lock to an operational position while user is roughly positioning the projectors. Middle and Right: Playback of a video after synchronization and registration on the setup of 4 projectors.***

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.

Algorithm 3 *Pseudo code for projecting devices in distributed scalable synchronization.*

```
Initialize display time adjustment to 0
if device is the synchronization reference then
    Set Sync flag to 1
else
    Set Sync flag to 0
end if
Read the internal time
Display the first synchronization frame which is a coded pattern containing device ID, sync flag and frame number 1
while not end of video playback do
    Wait for next display time based on reading internal time and applying time adjustment
    if finished synchronization sequence then
        show next decoded video frame
    else
        if display time adjustment is not zero then
            Set sync flag to 1
            Set display time adjustment to 0
        end if
        Show coded pattern for the next frame number
        if sync flag is zero then
            Capture n images equally spaced in time during the frame period and decode coded patterns
            if at least one of the neighbors has sync flag set in all n captured images then
                Select such synced neighbor as the reference
                Calculate display time adjustment comparing to reference as described in section 2.2
            end if
        end if
    end if
end while
```

We presented our method starting from introduction of a centralized synchronization scheme, then we described an approach to improve the accuracy of the synchronization to sub frame period level. Based on the same concept used in the centralized approach we introduced a network independent distributed synchronization algorithm. We extended this algorithm to a scalable distributed scheme to eliminate the requirement for each camera to cover the entire projection area. We implemented our method in a distributed network independent ensemble of tiled pico projector development kits to demonstrate the visual based synchronization in practice.

5. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Number ECCS-1059764. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

6. REFERENCES

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

[2] Texas instruments dlp1700 specifications. http://http://www.ti.com/product/dlp1700.

[3] 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.

[4] 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.

[5] S. Eilemann, M. Makhinya, and R. Pajarola. Equalizer: A scalable parallel rendering framework. *IEEE Transactions on Visualization and Computer Graphics*, 2009.

[6] A. Fai. Projected display shared workspaces. *US Patent Application 20110197147*, 2010.

[7] 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.

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

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

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

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

[12] P. Roman, M. Lazarov, and A. Majumder. A saclable distributed paradigm for multi-user interaction with tiled rear projection display walls. *IEEE TVCG*, 2010.

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

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