UNIVERSITY OF CALIFORNIA,
IRVINE

Resource Aggregation for Collaborative Projected Video from Multiple Mobile Devices

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical and Computer Engineering

by

Hung Nguyen

Dissertation Committee:
Professor Fadi J. Kurdahi, Chair
Professor Aditi Majumder, Co-Chair
Professor Ahmed M. Eltawil

2016

DEDICATION

To my Parents
who have always cared for me and believed in me in every instance

and

To my Spouse
who has supported me with unconditional love and patience in this long journey

and

To my Children
whose joy in learning gave me the greatest motivation to move forward

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

It was a long and hard road. More than once, I almost gave up. Most of the time, I thought I could never reach the end of such an arduous journey. Without the expert guidance and continued support of my Professors, I would have lost my way a long time ago.

Firstly, I would like to express my deepest appreciation to Professor Fadi Kurdahi, who consistently showed me how to find research problems in an era when most new ideas and technologies become obsolete very quickly. Professor Kurdahi led me in the right direction in carrying out research, encouraged me to overcome research bottlenecks and walked me step-by-step onto this research path.

Secondly, I would like to express my sincere appreciation to Professor Aditi Majumder who guided me as to how to write a good paper, one that would bring out problem details and their solutions. Professor Majumder also gave me a lot of precious technical advice and commentary during my researches.

My humble thanks to Professor Ahmed Eltawil for his advice in finding the better methods to verify the video images in the aggregated systems and providing an excellent lab environment to carry on my researches.

Without the invaluable support and assistance of my Professors, I would not have been able to go through these very first steps in my research journey.

CURRICULUM VITAE

Hung Nguyen

1990	B.S. in Electrical Engineering, HCMC University of Technology, Vietnam
2000	Advanced Software Technology Certificate, UCI Extension, University of California, Irvine, California
2003	M.S. in Electrical Engineering, California State University of Long Beach, Long Beach, California
2010	Embedded Software Certificate, UCI Extension, University of California, Irvine, California
2016	Ph.D. in Electrical and Computer Engineering, University of California, Irvine, California

1990-94	Researcher, Electrical Engineering Department, HCMC University of Technology, Vietnam
1995-07	Senior Design Engineer, General Electric Infrastructure, Costa Mesa, California
2007-08	Principal Software Engineer, Indigita Corporation, Irvine, California
2008-09	Senior Principal Software Engineer, Western Digital, Irvine, California
2009-16	Principal Design Engineer, Extron Electronics, Anaheim, California

FIELD OF STUDY

Resource Aggregation for Projector Enabled Mobile Devices

PUBLICATIONS

“Resource Aggregation for Collaborative Video from Multiple Projector enabled Mobile Devices”, ESTIMedia'16 Proceedings of the 14th ACM/IEEE Symposium on Embedded Systems for Real-Time Multimedia

“Rectangular Stable Power-Aware Mobile Projection on Planar Surfaces”, The 15th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI 2016, December 2016)

ABSTRACT OF THE DISSERTATION

Resource Aggregation for Collaborative Projected Video from Multiple Mobile Devices

By

Hung Nguyen

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Irvine, 2016

Professor Fadi Kurdahi, Chair
Professor Aditi Majumder, Co-Chair

We explore and develop an embedded real time system and associated algorithms that enable an aggregation of limited resource, low-quality, projection-enabled mobile devices to collaboratively produce a higher quality video stream for a superior viewing experience. Such a resource aggregation across multiple projector enabled devices can lead to a per unit resource savings while moving the cost to the aggregate.

The pico-projectors that are embedded in mobile devices such as cell phones have a much lower resolution and brightness than standard projectors. Tiling (putting the projection area of multiple projectors in a rectangular array overlapping them slightly around the boundary) and superimposing (putting the projection area of multiple projectors right on top of each other) multiples of such projectors, registered via automated registration through the cameras residing within those mobile devices, result in different ways of aggregating resources across these multiple devices. Evaluation of our proof-of-concept system shows significant improvement for each mobile device in two primary factors of bandwidth usage and power consumption when using a collaborative federation of projection-embedded mobile devices.

The portable, low-power, light weight, small size pico-projectors are key components of projection-enabled mobile devices for the future. Due to the reduction of weight and dimension and the portability nature of the projector-enabled mobile devices, the calibrated integrated systems are prone to physical un-stabilizing of the projected image during the presentation. Thus the auto re-calibration and projected video stabilization features during the presentation time becomes essential requirements to enhance user experience. The design, algorithm, and implementation methods for these features will be presented in the second part of the dissertation.

Chapter 1

Introduction

1.1 Challenges of Projector Enabled Mobile Devices in the future

Smart Mobile Device Users are over 2 billion in 2016

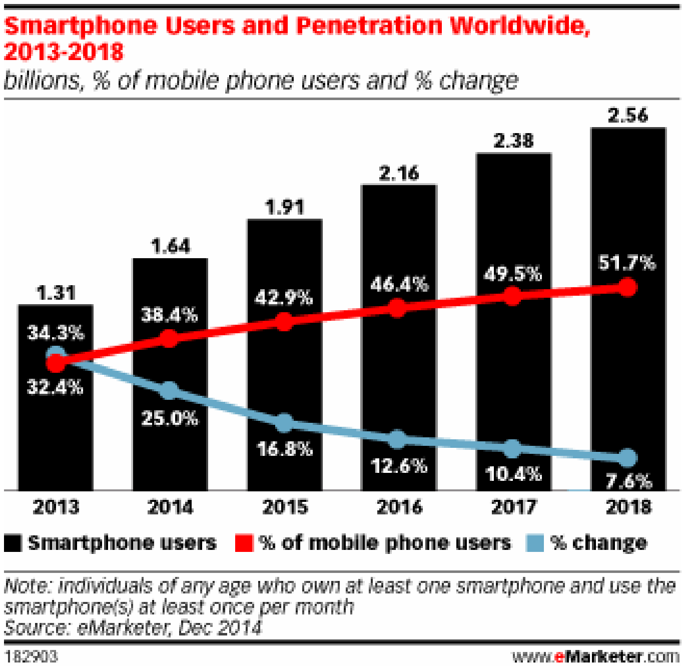
The number of smartphone users worldwide surpass 2 billion in 2016, according to new figures from eMarketer, Ref. [61]. In 2015, there will be over 1.91 billion smartphone users across the globe, a figure that will increase another 12.6% to near 2.16 billion in 2016 as shown in Figure 1.1. Mobile devices have transformed the way we live and conduct everyday activities. Not only can we access almost any type of content on mobile, but with most mobile smartphones today we can deposit checks, accept credit cards, order food and pay for groceries, sign digital documents, control the sprinkler, monitor the house temperature and even lock the house door. With many very convenient usages, as a result, accordingly to the estimation, the smart mobile users increase from about one-third of the global population in 2015, to over half by 2018, meaning that these smart features such as web accesses, video streaming, game, social interactions become more important if not essential user experiences nowadays.

Mobile Video Traffic takes about 75% of total huge traffic by 2010

Cisco's latest annual Visual Networking Index forecast (VNI), Ref. [62], states that mobile data traffic has grown 400-million-fold over the past 15 years, 3.7 exabytes per month in 2015, and expected to continue to grow over 30 exabytes per month by 2020. Also the increased number of users, mobile devices, and innovations in technology are driving mobile video consumption through the roof. Estimate by 2020, there will be more than 11.6 billion mobile connected devices and have a huge impact on video traffic as shown in Figure 1.2 with video will account for about 75% of that traffic. Not only that, 7 trillion video clips will be uploaded in 2020 - 2.5 daily video clips for every person that will drive a huge impact on the drive to have better social interaction environments for mobile device.

Limitation for user experience with small screen

Mobile devices with their small size have so convenient and portable characteristics. Compared with desktop and even laptop screens, phone screens accommodate a lot less content. As a result, small screen size is a serious limitation for mobile devices. The limitation of the small screen is even more severe with the higher resolution video contents have been offered that increase the user experience expectations for the mobile device display to carry out those contents.



Smartphone Users and Penetration Worldwide, 2013-2018

billions, % of mobile phone users and % change

Note: individuals of any age who own at least one smartphone and use the smartphone(s) at least once per month
Source: eMarketer, Dec 2014

182903 www.eMarketer.com

Figure 1.1 Smart Mobile Device Users are over 2 billion in 2016

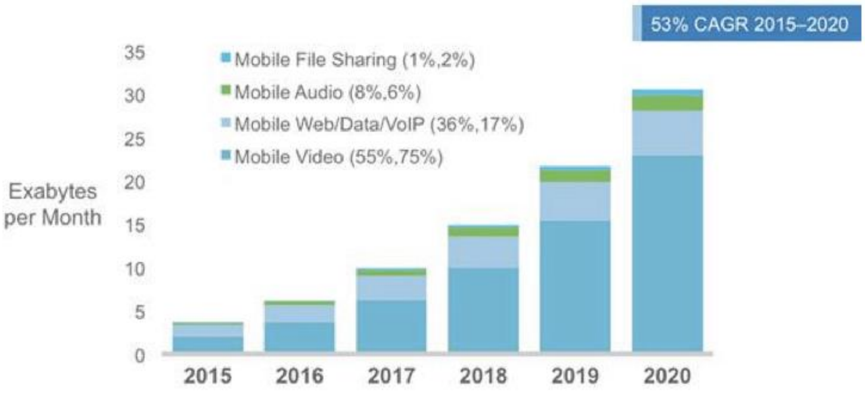
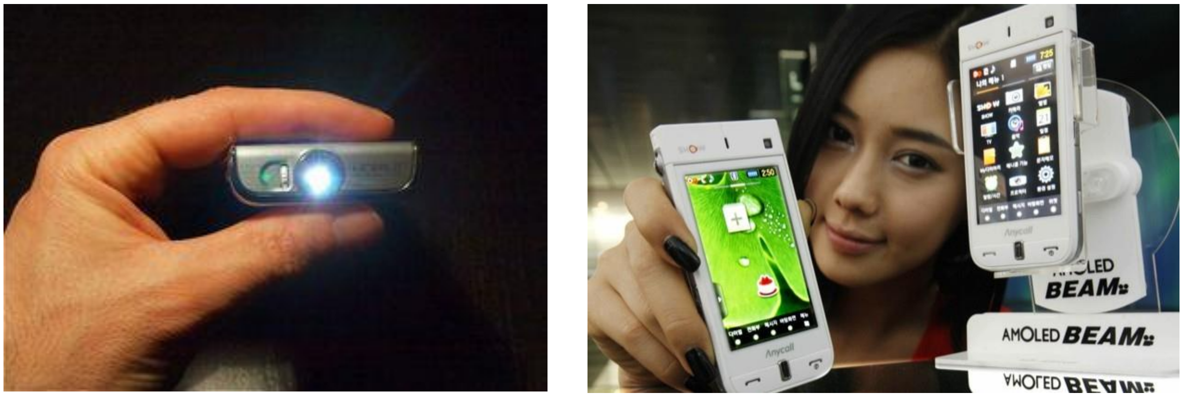


Figure 1.2 Huge mobile video traffic continues to rise in the next years

Moreover, sharing short video clips, quick presentation during social interactions are some immersed trends of user experiences that lead to the market opportunities to embed a pico projector into the mobile devices are huge. Such embedded projector phones will improve the usage of many applications and will lead to new interaction techniques. The projected display has the advantages of a larger size and potentially higher resolution when compared to the typical small screen of the mobile phone. However, there are still many open questions with regards to the effects imposed on the battery life of the mobile device. Secondly, the brightness of the embedded projector when compared to currently available projectors is also a concern.

As a result, embedded projector mobile devices in the market continue to grow although huge gap between higher user expectation and low brightness and resolution deliverability

The pico projector market is expected to reach USD 9.32 Billion by 2020, at a CAGR of 40% between 2014 and 2020. The growth of the market is propelled by the integration of pico projector in consumer electronics such as mobile phones and tablets. Pico projectors have undergone several technological advancements due to the continuous demand in the market. The pico projector comes in different types USB, embedded, media player, and standalone. Although standalone pico projectors account for the largest market share in terms of value at this moment; however, the market for embedded pico projectors is expected to grow at a high CAGR during the forecasted period. Beside Samsung (Korea) with its Samsung Show



(a) Samsung Show W7900 (b) Samsung Amoled Beam W9600

Figure 1.3 First Mobile Phones with a projector built in **(a)** Samsung Show W7900, projector resolution 480x320, 10 lumens, max image 50", released February 2009 (b) Samsung Amoled Beam W9600, projector resolution WVGA 854x480, 9 lumens, released April 2010

(a) Galaxy Beam I - i8530 (b) Galaxy Beam II - G3858

Figure 1.4 Android Samsung Galaxy Beam Mobile Phones with a projector built in
(a) Samsung Galaxy Beam I - i8530, projector resolution WVGA 800x480 pixels, GPU Mali-400, released July 2012 and (b) Samsung Galaxy Beam II, projector resolution WVGA 854x480, battery boost from 2000 to 2600mAh, equipped with accelerometer, gyro, proximity, compass sensors and a 5MPera, released April 2014

and Galaxy Beam I and II, Lenovo Group Ltd. (China), Sony Corporation (Japan), and Haier Group (China) manufactures also involved in the development and introduction of pico projectors into their products. The restraints for the development of pico projector include power consumption and low brightness of pico projectors are still the big concerns for this type of products.

Resource Aggregation solution for embedded projectors in mobile devices

A resource aggregation system that aims to bridge this gap between higher user expectation and lower quality imagery from embedded pico-projectors by aggregating resources from multiple devices in a federation of projection-enabled mobile devices which can deliver a better quality viewing experience (in terms of resolution, size, brightness, or frame rate). Due to the gap between user expectation and quality delivered by projector-enabled mobile devices, it is obvious that the techniques to tile or superimpose projections from multiple devices, thereby increasing their size/resolution and brightness/frame rate, respectively. In other words, this opens up the possibility of aggregating the display resources across multiple devices to create a superior viewing experience than what can be offered by a single device.

1.2 Resource Aggregation - Power Saving Issues and Solutions

We will explore and develop an embedded real time system and associated algorithms that enable an aggregation of limited resource, low-quality, projection-enabled mobile devices to collaboratively produce a higher quality video stream for a superior viewing experience. Such a resource aggregation across multiple projector enabled devices can lead to a per unit resource savings while moving the cost to the aggregate. The pico-projectors that are embedded in mobile devices such as cell phones have a much lower resolution and brightness than standard projectors. Tiling (putting the projection area of multiple projectors in a rectangular array overlapping them slightly around the boundary) and superimposing (putting the projection area of multiple projectors right on top of each other) multiples of such projectors, registered via automated registration through the cameras residing within those mobile devices, result in different ways of aggregating resources across these multiple devices. Evaluation of our proof-of-concept system shows significant improvement for each mobile device in two primary factors of bandwidth usage and power consumption when using a collaborative federation of projection-embedded mobile devices. This is the first time a real-time solution for aggregation of resources across a federation of low-cost, low power mobile devices can be achieved completely and automatically that can result in a viewing experience of as high as 4K (3840x2160) content with integrated four mobile devices playing 1080p content.

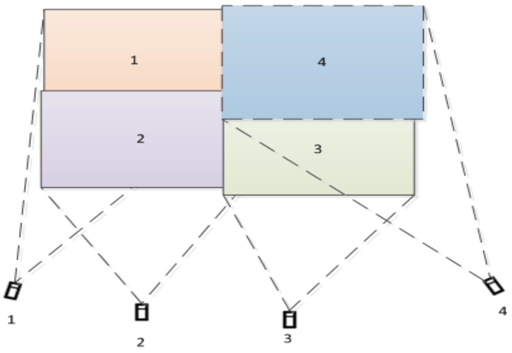


Figure 1.5 Tiled setting – Aggregate Resource (Power, Bandwidth)
Higher resolution than single device

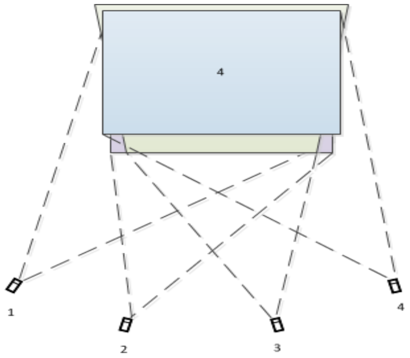


Figure 1.6 Super Imposed Setting
Higher brightness than single device

1.3 Auto Re-Calibration in Federation Setting

Due to the mobility and portable features of a mobile device, the misalignment incidents during playing is substantial. This issue drives to the auto re-calibration needs to detect and correct that misalignment during playing.

1.4 Self-adjusting Projector in Real Time System

One main advantage of pico-projector enabled mobile devices is its portability and ability to project video anywhere and anytime suitable for social events, quick and small-sized presentations as well as entertainment. However, one major issue in using these hand-held devices is the instability of the video display during presentation. In Chapter 5, we present two embedded real time systems and associated algorithms that allow stabilizing the video during the presentation to enhance user experience. The first system uses a camera to detect and correct the displacement in real time. The camera takes the projected video and measures the difference from the reference frame location parameter to the new frame location parameter, compares it and performs the compensation seamlessly during video playing. The second system employs an Inertial Measurement Unit (IMU) to estimate the rotation and transformation of the mobile device and compensates the video frames during playback accordingly. While the camera-based visualized feedback system can compensate more accurately over a long time period due to its closed-loop feedback characteristic, the IMU system can respond faster based on its response time and open-loop control system. Furthermore, the camera-based system requires a fixed marker which may be impractical in many settings. To the best of our knowledge, this is the first system demonstrating the stabilization of projector enabled mobile video in real time with video compensation performed entirely by video processing.

Chapter 2

Integration System of Multiple Projector Enabled Mobile Devices

2.1 Introduction

Multi-projector displays have been the mainstay of large data visualization systems more than a decade now and there is a rich body of literature in the visualization community on how to register them automatically using cameras, both geometrically, Ref. [1], [3-7], [9-11] and photometrically, Ref. [8]. Federation of mobile projectors was first explored in Ref. [2]. Although the automated registration of the aforementioned enabled such thinking about federations, temporal synchronization is of low priority in LAN based visualization systems. Ref. [2] presented camera-based automated techniques for temporally synchronizing such federations. Using a visual interface via a camera for temporal synchronization is a critical capability in the context of mobile devices to bypass wireless networks fraught with congestion and delays. However, since almost all the processes are done in CPU, this suffers from low frame rate that cannot achieve video streaming applications which demand real-time performance.

In this Chapter, we present the complete embedded video streaming system using federations of projector enabled mobile devices in real time. This entails performing all the tasks of camera-based geometric and temporal registrations performed using the embedded hardware in real time using several critical improvements described in detail in later sections. These operations enabled us to perform the first set of extensive experiments on how such federations can be used for aggregating resources across projection based devices in the context of video streaming and derive benefit from it. We show that this can lead to significant power and bandwidth savings even with relatively simple streaming methodologies.

2.2 System Overview

Our setup consists of multiple tiled or superimposed pico projectors each connected to an embedded system with a camera to capture the projected video. The embedded system loads the video content from its local storage or from the Ethernet server. It then decompresses the video content and plays its spatially segmented video. In order to collaboratively tile together the video, each embedded system performs the registration, video synchronization, and video transformation processes before projecting the video to its HDMI port.

Herein we describe the system in detail:

First, we list the three components in each projector embedded mobile device. We call it a mobile plug and play projector (MPPP)

1. The first component is a low-cost Raspberry Pi version B board, Ref. [13], shown in Figure 2.1, that contains the Broadcom SoC BCM2835, a high definition 1080p Embedded Multimedia Applications Processor. This SoC contains one low power ARM1176JZ-F Applications Processor (CPU) and a dual core VideoCore IV® Multimedia Co-Processor (GPU) with 1080p30 Full HD High Profile H.264 Video Encode/Decode engines. This component is a typical low cost, low power, low resource hardware platform was chosen as a processing unit similar to a typical low-cost mobile device platform. It also was chosen for our prototype to have popular open source software accesses for experiments and applications.
2. The second component is a low resolution pico projector from Texas Instruments, Ref. [12], shown in Figure 2.2, a DLP1700 kit with HVGA 480x320 resolution, DVI-D input and VGA 50/60Hz format, serving for our low resolution aggregation demonstration.
3. The last component is a typical low-cost camera that has a five mega pixel resolution, and supports up to 1080p 30Hz, 720p 60Hz, or VGA resolution with a refresh rate of up to 90Hz, shown in Figure 2.3.

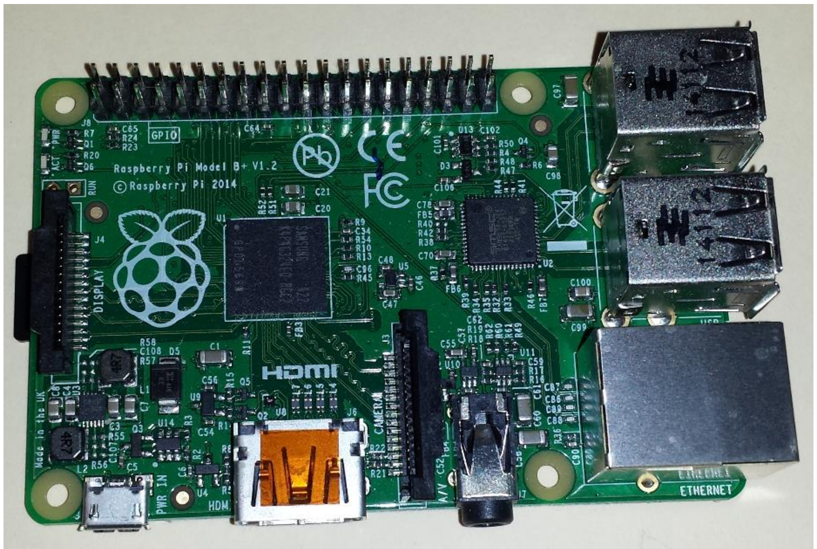


Figure 2.1   First component is a low cost Raspberry Pi version B

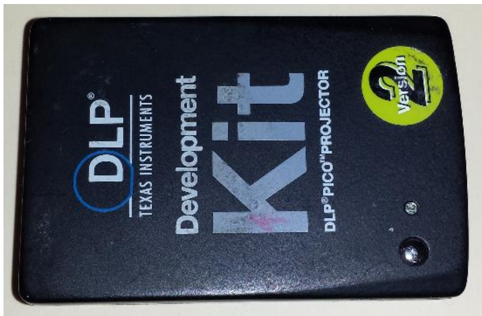


Figure 2.2   Second component is a low resolution pico projector

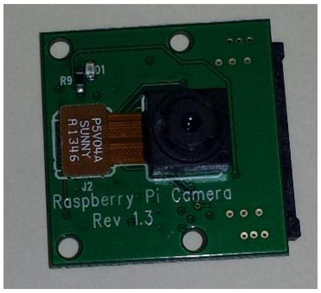


Figure 2.3   Third component is a low cost 5 Mega pixel camera

Using this three-component MPPP unit as a projector-enabled mobile device model, we built a prototype systems consists of two MPPP units that collaborate in a loose physical setting. That one consists of two sets of pico projector systems with embedded camera Raspberry Pi boards to perform the collaborative federation. As shown in Figure 2.4, these two projectors are tiled to create a simple collaborative federation system which can perform fast registration and synchronization and can be used practically in most loose environment settings. The same prototype has also been used to develop and verify our real time processes quickly.

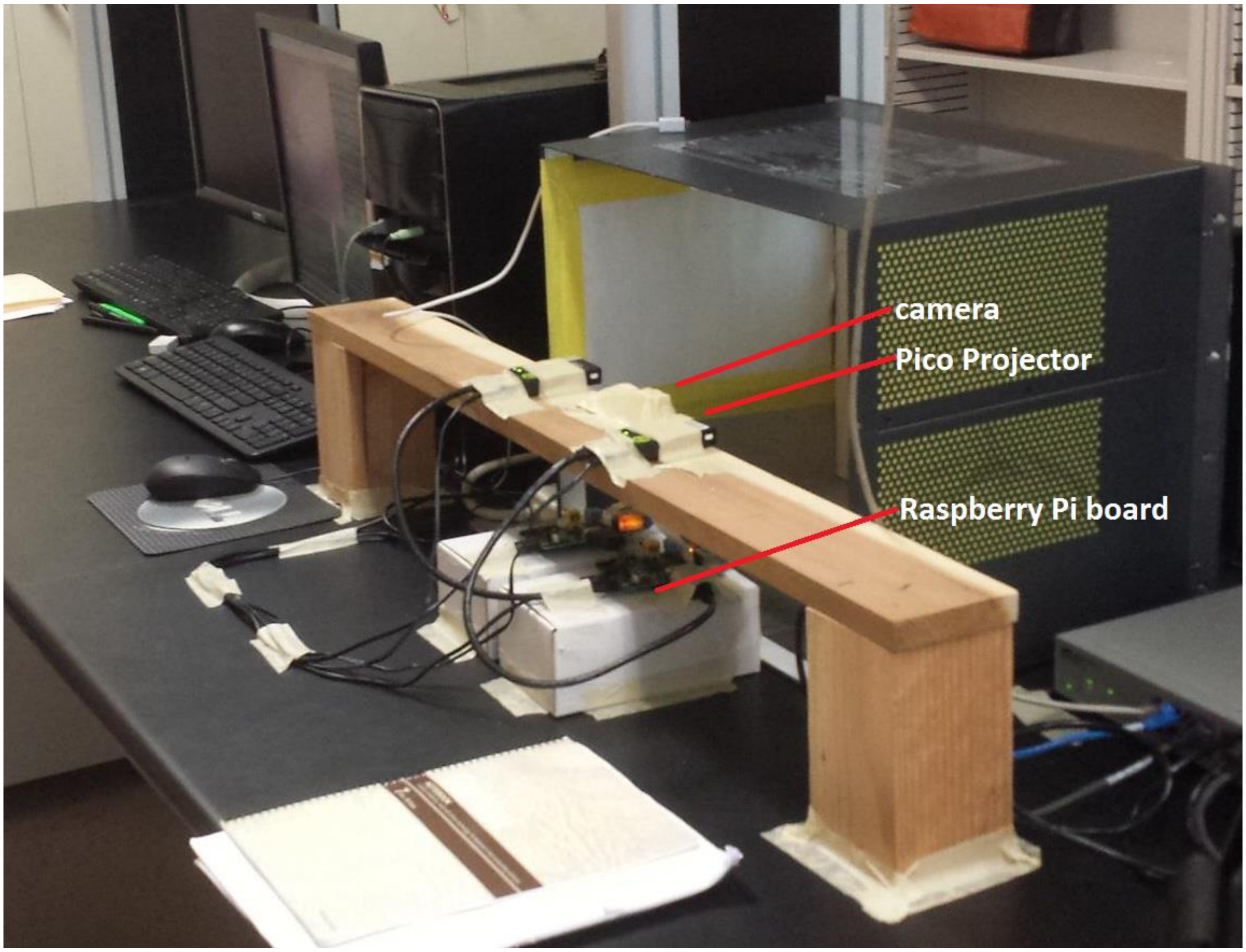


Figure 2.4 Two MPPP units aggregate in the first two unit prototype model

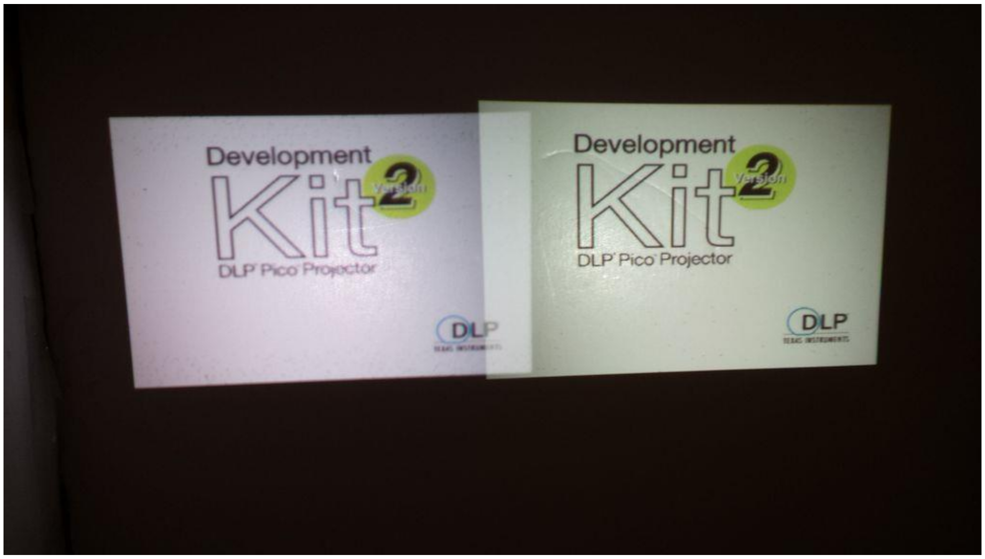


Figure 2.5 Two MPPP units at start up before calibration in the tiled setting

The second prototype was where four units were tiled together as shown in Figure 3.12 to create the federation. Note that the devices had to overlap each other to avoid getting discontinuous with slight movements.

Since the camera field of view is usually much wider than that of the projector, we assume that the entire projected image can be seen by all the cameras on the mobile devices. One of these cameras can be used to achieve the geometric and temporal registration as described in Ref. [2]. These are achieved by QR based codes embedded in the projections which are seen by the camera. More details are available in Ref. [2]. Following this one time calibration process, these parameters are then used in run time to correct and synchronize in real time every frame of the video that is displayed. When playing the video following this calibration, every frame needs to be warped to be registered geometrically, attenuated to be registered photometrically and synchronized in real time. This has to be achieved on the compressed video content instead of playing each raw frame individually. In order to achieve it, we exploit the GPU engine to decode the h264 video, then render the decoded video to the graphic layer where both blending process and warping pixel manipulation is done in real time. Note that the system uses pre-split videos - either resolution split in tile setting or frame rate split in superimposed setting - for reducing bandwidth data usage and power consumption for decoding and rendering on each individual endpoint unit in these aggregation settings. Figure 2.5 shows two tiled units at start up time before performing the calibration process. Figures 2.6 and 2.7 show the snapshot of a seamless video thus produced from the multiple tiled and superimposed units respectively after performing calibration.

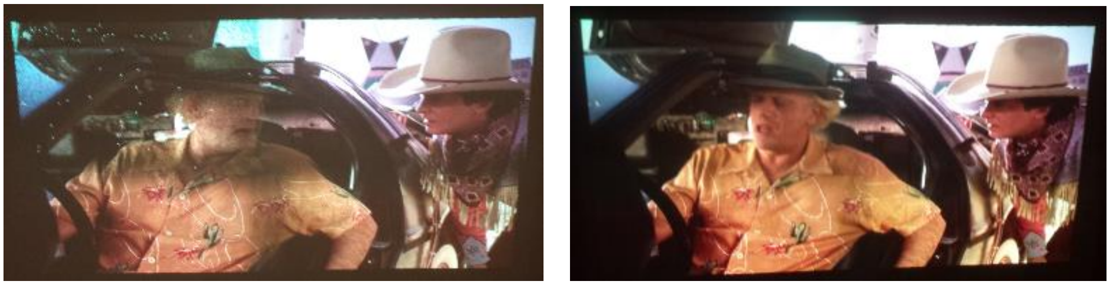


Figure 2.6   Tiled setting image - before (left) and after (right) calibration

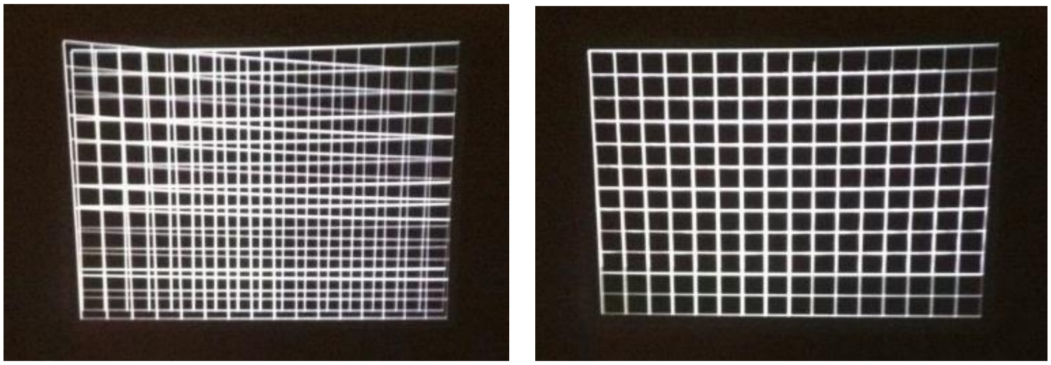


Figure 2.7   Superimposed setting image – before (left) and after (right) calibration

The block diagram for the methods in each of these prototype systems is shown in Figure 2.8 for both tiled setting and superimposed setting. Each participating unit starts with the initiation stage with yellow blocks in this block diagram which achieves the geometric and temporal synchronization. In this stage, the CPU loads the QR codes to be used for camera-based registration from its media storage SD card and projects them sequentially.

Simultaneously, the CPU captures the whole projected image with its embedded camera. The frame data, which was embedded in QR codes, is extracted from each captured image. The CPU also graphically detects the corners of the QR codes and the centers of the blobs embedded in them from the captured images. It continues the capture and extract processes until the desired data can be collected, validated, and used. This desired data contains two sets of information: the set of coordinates for features embedded in the QR codes and the set of frame number parameters. These two sets of information are then used to output parameters that can achieve two objectives, namely (a) the mathematical parameters (in the form of a matrix called homography) that will be used by each of the units to warp their images so that the images from the multiple unit are registered or stitched together; (b) the blending parameters (dependent on the amount of overlap across the projectors) so that the colors from the different projectors are blended in the overlap regions to create a single display; and (c) the delays that each unit should apply to be temporally synchronized with each other. These parameters are then used in the next step, the real time PRESENTATION stage. Figure 2.8 shows the block diagram for the entire pipeline. Its block will be presented in more detail in the following section.

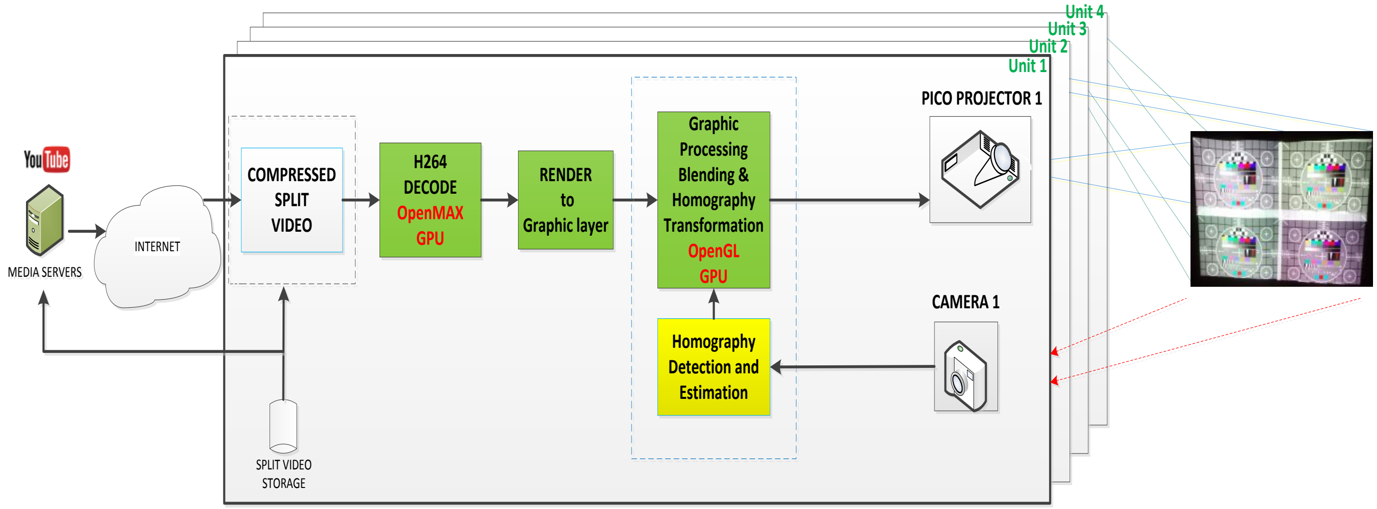


Figure 2.8 Block Diagram of Collaborative Projector enabled Mobile Devices

2.3 Link from Calibration to Presentation Stage for Real Time Performance

Since the superimposed settings are constructed with similar functional blocks as of the tiled setting, throughout this chapter we describe only one system for both settings. Due to the physical geometric differences of projected video positions, there are parameter differences in each functional block such as in video transformation block and homography estimation block. These different parameters had to be taken care of when constructing the superimposed setting from the tiled setting. In addition, there is one process, - slot searching, - which is the major additional concept for the superimposed setting. In the superimposed setting, since its QR codes are projected to an identical location, the QR codes are unable to detect correctly due to the overlapping interferences among them. In addition to the initiation processes for both tiled and superimposed settings, we need to find a reliable algorithm to overcome the overlapping interferences of its projected QR codes in this initialization stage. To detect them separately, we search the vacant and allocate the unique time slot for each unit to project its QR codes. The time slot searching process is performed at the beginning of the initiation stage in the superimposed setting system.

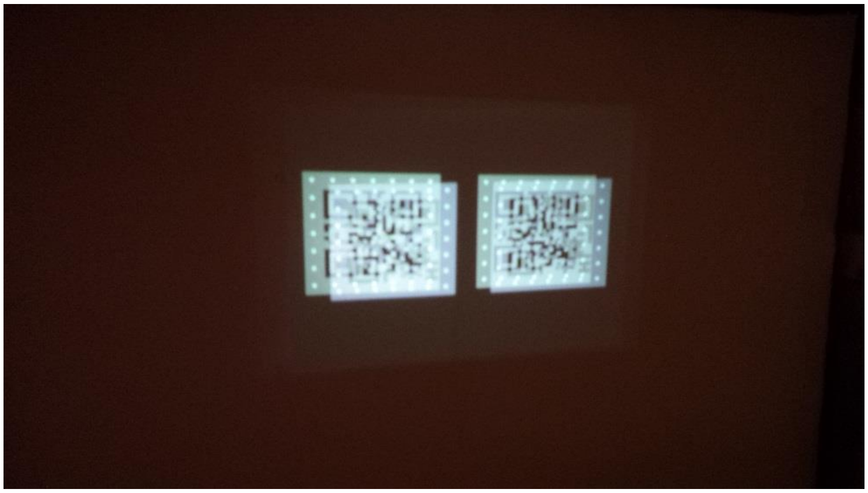


Figure 2.9 Superimposed setting – during searching slot – QR overlapped

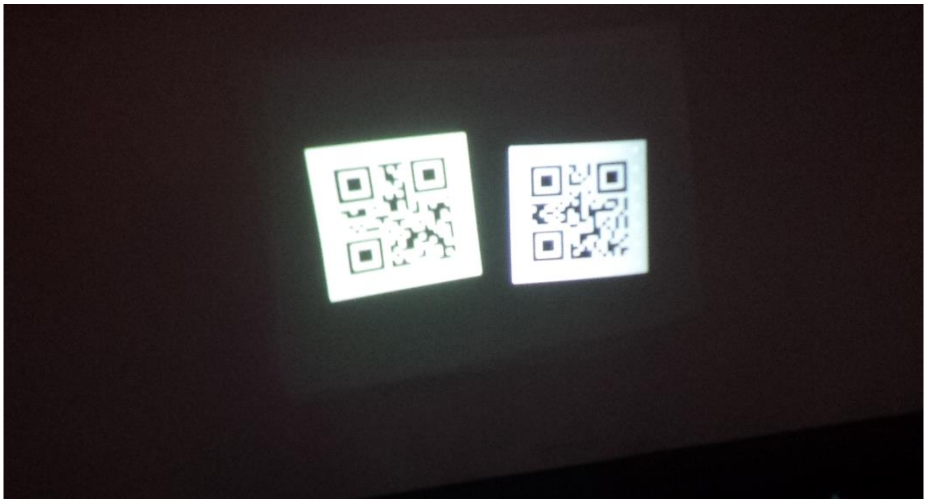
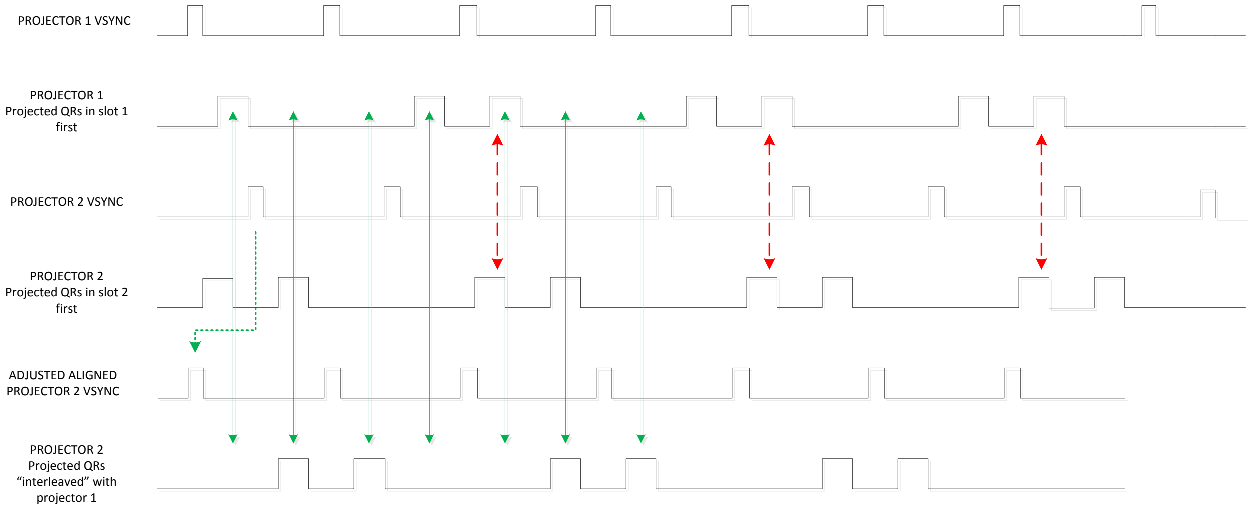


Figure 2.10 Superimposed setting – searching slot completed and slots were allocated – each QR projected from each projector sequentially


PROJECTOR 1 VSYNC
PROJECTOR 1 Projected QRs in slot 1 first
PROJECTOR 2 VSYNC
PROJECTOR 2 Projected QRs in slot 2 first
ADJUSTED ALIGNED PROJECTOR 2 VSYNC
PROJECTOR 2 Projected QRs “interleaved” with projector 1

NEW FIND SLOT – TWO PROJECTORS MODEL

Note:
1. dash red line: intermittently overlapped
2. dash green line: projector 2 delayed each time step eventually getting aligned with VSYNC of projector 1
3: solid green line: QRs are now projected in an interleaved way

Figure 2.11 Superimposed setting – searching vacant slot process

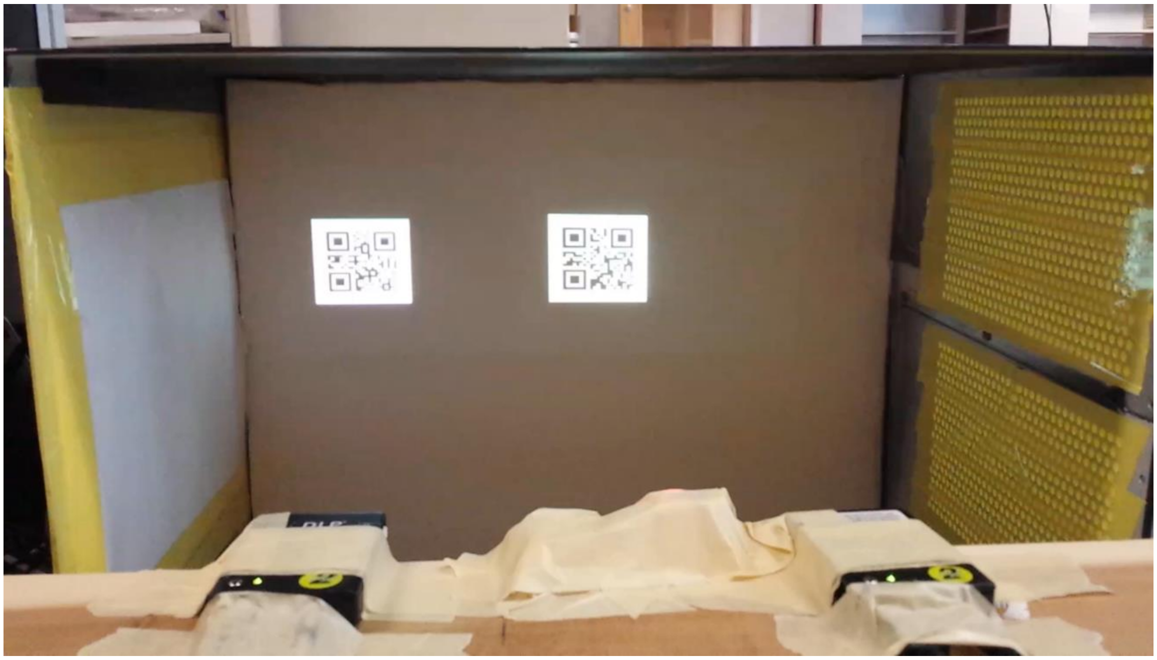


Figure 2.12 Aggregated projectors in a loose setting environment

Then the initiation processes are the same for both settings as follows:

The yellow blocks in Figure 2.13 show the components of the initiation stage. In this stage, each unit loads the QR codes that contain their projector identification and frame number encoded as a QR code pattern. Next, the unit projects a sequence of frames at a targeted frame rate, i.e. 30 fps. Simultaneously, the capture process is started to secure the whole image that includes all of projected QR codes from neighboring projectors. The captured image is processed to detect the positions of features (corners and blobs) on the QR codes. The features on each frame are used to estimate the homography using linear regression techniques. The homography matrix parameters, thus calculated, are then transferred to the video transformation process.

Simultaneously, for video synchronization, the QR code validation and extraction processes also extract the frame number with the projector identification, accordingly, to determine the frame delay of each projector. The frame delay of a projector is the difference between its frame number and the smallest frame number in the group. This frame delay is transferred to the video synchronization process to offset the frame delay in the next PRESENTATION stage.

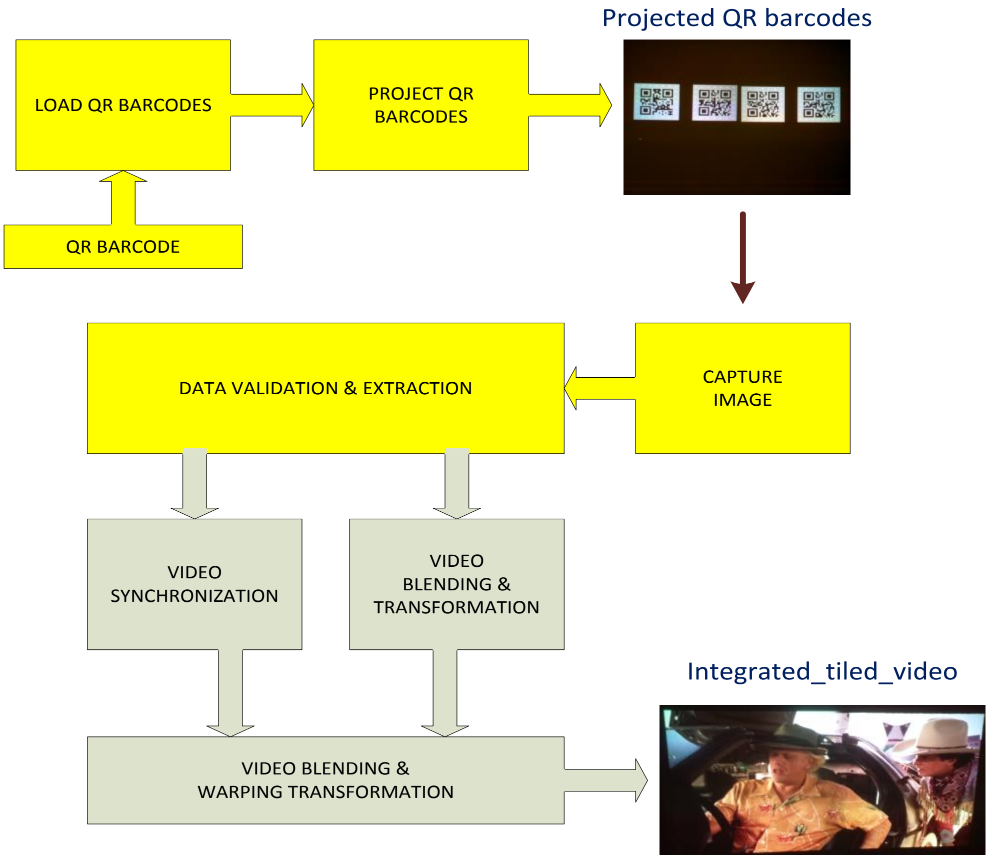


Figure 2.13 Calibration and Presentation Stages Perform as Continuity Back-to-back Processes

2.4 Video Synchronization Overview

The frame delay number data that was detected and calculated in the initiation stage will be used in the video synchronization process right before video presentation. After the first initiation, the different frame number is used to count back the number of frames the unit needs to wait. During the waiting time, the unit still projects the QR codes for other units to continue getting the samples if they have not finished their first stage. Figure 2.14 shows four projectors projecting the QR codes with frame number increments, sequentially. Each unit captures the images, detects the frame number of each of the units, and calculates the frame delay which is the difference between the frame number itself and the most lagging unit. Then it waits for its frame delay value before continuing to project the next frame number until their frame numbers reach the designed maximum frame. Thus, the number of frame delays will offset the time lagging among users.

Let M be the designed maximum frame, t be the desired time lagging allowable and R be the number of projected frames per second. In order to have a successful synchronization process, this equation has to be satisfied:

$$M \geq Rt \tag{2.1}$$

To accommodate the high rate and the reasonable time lagging among users, the number of frames M has to be large enough. However, when the playing rate is low, the waiting time is very substantial. We reduce the waiting time by changing the value of M with the rate time accordingly to prevent waiting time in the synchronization process.

Another improvement in the synchronization process is eliminating the time variation in the video loading and preparing processes of different units at different times. First, the video content is loaded then paused. Second, frame synchronization performs. Finally, the video is un-paused to start playing. These steps can be done with a video buffer control. This method was verified to eliminate the delay variation in both of our cases - playing video content from local storage and from remote internet servers.

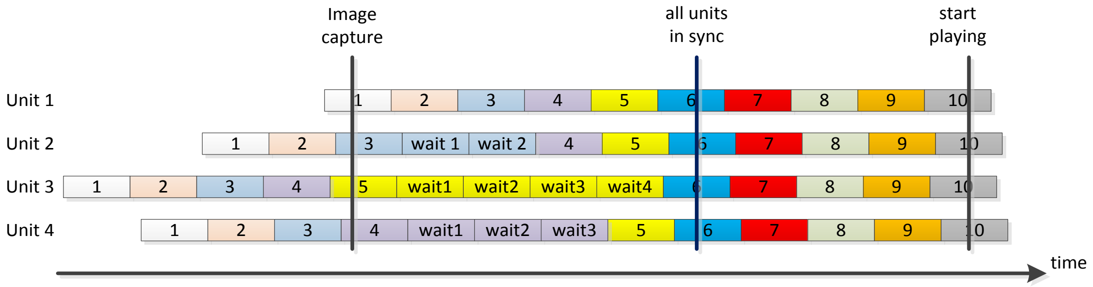


Image capture
all units in sync
start playing
Unit 1
1 2 3 4 5 6 7 8 9 10
Unit 2
1 2 3 wait 1 wait 2 4 5 6 7 8 9 10
Unit 3
1 2 3 4 5 wait1 wait2 wait3 wait4 6 7 8 9 10
Unit 4
1 2 3 4 wait1 wait2 wait3 5 6 7 8 9 10
time

Figure 2.14 Video Synchronization Process with a Four-Projector Model

2.5 Registration Overview

One major difference between the multiple-projector displays and video wall applications is geometric distortions. There are two types: intra–projection and inter-projection distortions Ref. [6]. Intra-projection distortions are caused by off-axis projection with the assumption of playing on planar surfaces. Inter-projection distortions are due to unaligned edge boundaries. Both types of distortions are addressed using our camera-based registration techniques. Estimated homography is used to correct the multiple-projector video display with homography transformation.

Let's consider *N* feature points in the QR codes. Let the known coordinates of the *ith* feature point, *1 <= i <=N* in the projector's coordinate system be *(x_i, y_i)*. Let the corresponding point detected in the camera's coordinates by *(Xi, Yi)*. These two corresponding points are related by

$$\begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \tag{2.2}$$

When Equation (2.2) is converted to a linear equation, we get:

$$\begin{pmatrix} -x_i & -y_i & -1 & 0 & 0 & 0 & x_iX_i & y_iX_i & X_i \\ 0 & 0 & 0 & -x_i & -y_i & -1 & x_iY_i & y_iY_i & Y_i \end{pmatrix} (H)^T = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{2.3}$$

When all the *N* points are written out in this form, we get the matrix $AH^T = null$ vector when A is a 2Nx9 matrix. Usually, homography matrix has eight degrees of freedom, and hence h_9 is assumed to be 1. Therefore, we have to estimate the eight unknowns of H from this equation. Note that since we have a large number of features, this leads to an overly constrained system. Therefore, we find the values of H using a linear least square that minimizes the L2 norm given $\|Hx - X\|$. For the inter-projection distortion correction as mentioned earlier, we calculate the final homography H_i'set for each projector as follows:

$$H_i' = H_i^{-1} H_1 O_i \tag{2.4}$$

H_1 is the homography of the reference frame, which was chosen from the first projector in the four-projector model. To be generalized in the larger federation scale, this is the first set of each of four projectors in the neighborhood.

H_i^{-1} is the inverse transformation of matrix H_i.

O_i is the offset position matrix of projector *ith* in the four-projector or two-projector model. This offset position matrix is a unity matrix in the superimposed setting but in the tile setting, it has an offset value in the last column in accordance with the physical position of the projected video *ith*.

2.6 Video Transformation Overview

One of the major issues that we had to overcome is the capability to play and manipulate the compressed video simultaneously. In order to stream the video at 30 fps, we need to play and perform the video transformations, i.e. homography transformation and overlap blending with the compressed video content instead of playing each raw frame individually. After the compressed video content has been decoded by the GPU, the video stream is transferred to the video transformation processes. Since openGL ES, Ref. [15], supports graphic manipulations on Raspberry Pi, instead of rendering to HDMI video out by OpenMAX, Ref. [14], we rendered it to graphic layer to make use of openGL to do the graphic processing. Thus a common integration path between openMAX API and openGL ES on Raspberry Pi was established to link them together in our application software.

The blending process is used to mix individual partial videos at their overlapped areas. The edge blending for projected videos is a challenging issue for low cost embedded systems, especially when playing higher refresh rate video contents, i.e. 30 fps. We exploit the GPU engine in the Raspberry Pi to decode the h264 video then render the decoded video to the graphic layer where both a blending process and pixel manipulation for homography transformation is done in real time. Figure 2.15 shows the simple graphic processing flow to analyze the timing budget. The H264 Decode engine completed its decoding task less than 30ms for each frame. This first stage meets the timing requirement to be less than the frame period which is 33ms. For the second stage that includes two processing blocks, render to graphic layer and graphic processing, the video transformation and blending processes take longest time to perform and are the bottleneck of CPU-only core system type to meet this timing budget 33ms. Raspberry Pi has an advantage with its GPU engine working simultaneously with the CPU engine, handles the graphic process in much less time. For example, the transfer data and blending processes takes about 11ms in total and video transformation processes takes about 7ms for each frame thus meets the timing budget for this second stage of the pipeline. Thus by doing the homography estimation in the initiation stage and the homography transformation and blending using GPUs in real time, we have achieved real time operations of 30fps.

Since the video processing speed limitations of the CPU-only core architecture, the video processing tasks cannot be handled simultaneously. The process of calibration and playing tasks cannot be done on the fly. Also, the resolution has to be limited to HVGA for the video processing time interval to catch up with the frame rate. Thus we explored the opportunity with the combined co-working GPU and CPU architecture in the inexpensive SoC, such as the Raspberry Pi prototype board.

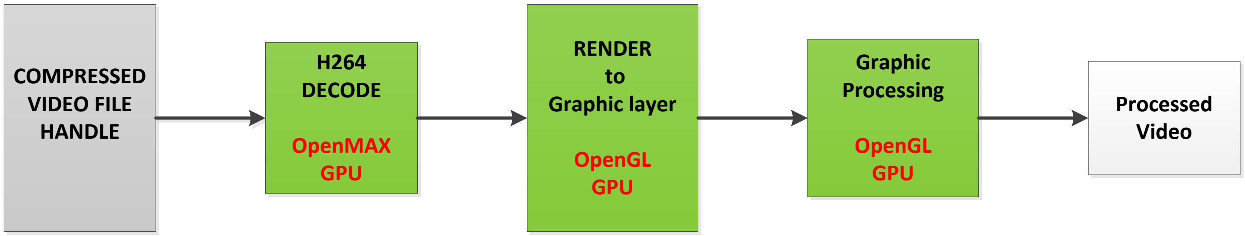


Figure 2.15 Simple Graphic Processing Flow for Video Transformation

Chapter 3

Resource Aggregation

3.1 Introduction

Portable, low-power pico-projectors are key components of projection-enabled mobile devices for the future. Such a projector consumes about 1.5 watts and weighs around 4-5 ounces compared to about 300 watts of power consumption and 7-11 pounds in weight of a standard presentation projector, Ref. [2]. The weight and dimension of a pico-projector can be reduced even further once embedded into a mobile device. This unprecedented reduction in power consumption and weight makes embedding pico-projectors in mobile devices practically viable. However, the portability and power efficiency features of pico-projectors come at the cost of a severely reduced image quality in terms of brightness and resolution. This creates a huge gap between user expectations and deliverability of video quality for projection-enabled mobile devices.

In this Chapter, we present a resource aggregation system that aims to bridge this gap between higher user expectation and lower quality imagery from embedded pico-projectors by aggregating resources from multiple devices in a federation of projection-enabled mobile devices which can deliver a better quality viewing experience (in terms of resolution, size, brightness, or frame rate). Interestingly, we show that contrary to intuition, this enhanced viewer experience also leads to resource (power or bandwidth) savings for each user participating in the federation, shifting the cost to the aggregate resources used from the service provider. This leads to a situation where users can realize a better viewing experience while saving resources on each of their devices. We show that using this paradigm of aggregated resources, it is possible to provide a 4K UHD viewing experience from mobile devices with only 300mW of power usage which takes only 14% of system resources per device, a feat that is currently impossible.

3.1.1 Main Contribution

In this Chapter, we propose a low-cost, low-power, embedded system that collaboratively aggregates resources from multiple pico-projectors in a tiled and superimposed setting. The goal is to enhance the user viewing experience by either increasing the video resolution/size by tiling, or the brightness/frame rate by superimposing. We present a video processing method by exploiting the GPU engine in the low-cost embedded system to perform the entire aggregation process in real time. The compressed video can be split and played from a local storage or from the Ethernet servers. We evaluate the resource savings resulting from this collaborative federation setting by focusing on the two most important resources of power and bandwidth. We show empirically that these savings can be significant for each device by experimenting with a large database of 700 videos divided into 10 different categories. We show that this savings comes at the cost of increased aggregate resources which we show can be mitigated by the extra power savings via *dynamic voltage frequency scaling* that exploits the very fact that each unit provides a lower quality video (in terms of resolution or frame rate), Ref. [60].

3.1.2 Prior Works

Multi-projector displays have been the mainstay of large data visualization systems for more than a decade now, and there is a rich body of literature in the visualization community on how to register them automatically using cameras, both geometrically as mentioned in section 2.1. These works address the issues of stitching the images from multiple projectors to make them look like a single display when using standard projectors and based desktop engines. Portability, power and bandwidth consumption issues on low-cost, limited resource engines like mobile devices have not been explored in this domain yet. Due to the gap between user expectation and quality delivered by projector-enabled mobile devices, it is obvious that the techniques in the aforementioned works can be used to tile or superimpose projections from multiple devices, thereby increasing their size/resolution and brightness/frame rate, respectively. In other words, this opens up the possibility of aggregating the display resources across multiple devices to create a superior viewing experience than what can be offered by a single device.

When considering automatic registration of such federations in LAN based visualization systems, temporal synchronization has been a low-priority concern since the maximum lag without any synchronization is less than 2 frames which is often imperceptible given the large field of view that such displays occupy, Ref. [16]. However, this becomes a severe bottleneck for mobile devices due to network congestion. An initial attempt was made for synchronizing a federation of pico-projectors temporally, Ref. [2], using the visual interface offered by an external camera, thereby bypassing the wireless network fraught with congestion and delays.

In this Chapter, we use all these federations of superimposed and tiled mobile projector settings mentioned in Chapter 2, and focus on studying the usage of power and bandwidth resources in such federations. We study how these stringent resources can best be utilized when federations of mobile devices are employed. In particular, we explore if rationing resources in a different manner can aid in creating a much superior viewing experience.

3.2 Resource Usage vs. Projected Video Quality

The primary motivation behind a federation of projectors is to explore the improvement in quality or user experience that can be achieved. For example, tiling 4 projectors allows the users to view the video at a better quality of 4 times the size or resolution. Alternatively, superimposing 2 projectors allows the users to experience the video at twice the brightness.

The tiled setup of projectors improves viewing experience of the users in terms of the resolution and the size of the projected video frames. Furthermore by extending the size of images it allows for reduction of the throw distance of projectors trading size off for more brightness. The relation between the viewing parameters in a collaborative tiled setting can be described by the following equations, intuitively:

$$R_n = N * R_0 * k \tag{3.1}$$

$$S_n = \left(\frac{d}{d_0}\right)^2 * N * S_0 * k \tag{3.2}$$

$$B_n = \left(\frac{d_0}{d}\right)^2 * B_0 \tag{3.3}$$

N : number of tiled projectors

R_0 : Resolution of video projected by a single projector

R_n : Resolution of video projected by tiled N projectors

S_0 : Size of projected image by a single projector at a throw distant d_0

B_0 : Brightness of projected image by a single projector at a throw distant d_0

S_n : Size of projected image by N projectors at current setting throw distant d

B_n : Brightness of projected image by N projectors at current setting throw distant d

k : overlapped factor, about 0.95-0.97

Let brightness ratio between tiled N projectors and single projector be

$$b = \left(\frac{B_n}{B_0}\right) \tag{3.4}$$

And resolution ratio be

$$r = \left(\frac{R_n}{R_0}\right) \tag{3.5}$$

And size ratio be

$$s = \left(\frac{S_n}{S_0}\right) \tag{3.6}$$

The relationship among these ratio parameters from equations (3.1)-(3.3) is represented by this equation:

$$r = b * s \tag{3.7}$$

Figure 3.1 shows the trade-off of these parameters in this equation. The brightness will be increase as the throw distance is decreased, or the projected image size is decreased, or the video resolution is increased. Note that from this relationship, the tiled setting is not only used to increase the resolution but also to increase the brightness thus enhance the users' experiences in both resolution and brightness aspects per their preferences.

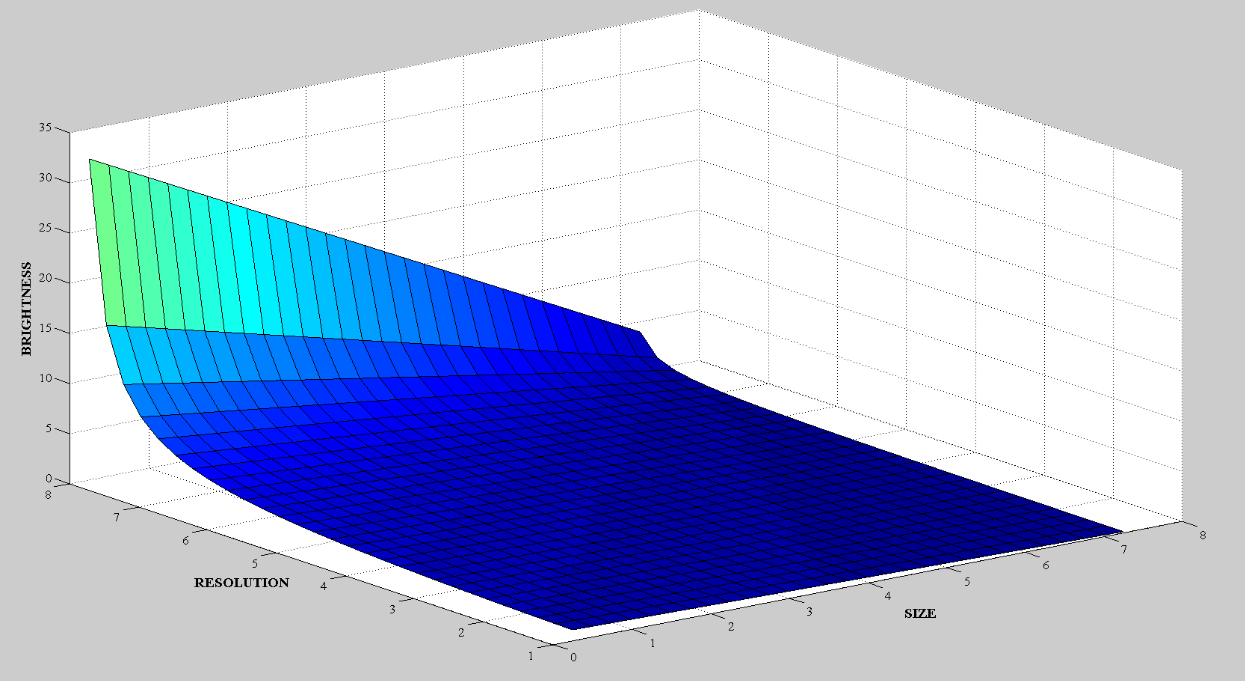


Figure 3.1 Trade-off between Brightness and Size as a function of throw distance for tiled projectors – Aggregated tiled projectors indicate increased resolution can improve the brightness while keeping the same projected image size

However, the question that becomes imperative in this context is the cost, if any, we are paying in terms of resources. In this section we explore this resource usage vs. quality tradeoff.

To achieve this we performed several experiments on a large number of videos. We collected 70 video samples in each of 10 different *categories:* static image, trailers, nature, people, shows, news, crowds, sports, music and animation as shown in Table 3.1 below. We designed these categories based on the most popular content that people usually watch.

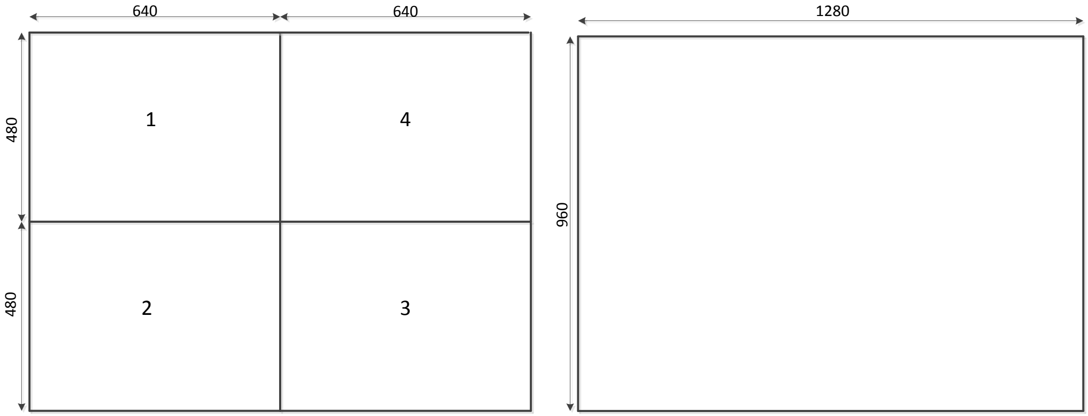
Then we report our findings from multiple experiments and measurements that we performed to analyze the trade-offs between the resources used and the viewing experience.

Table 3.1 Characteristics of Different Video Categories

#	Category	Resolution	Frame Rate (fps)	Average total bit rate (kbps)
1	animated	1280x720	23/24/25/29/30	1901
2	crowd	1280x720	29	3332
3	music	1280x720	24/25/29	3444
4	nature	1280x720	23/29	2350
5	news	1280x720	29/30	2222
6	people	1280x720	25/29	2383
7	shows	1280x720	29	1718
8	sports	1280x720	24/25/29	3862
9	static	1280x720	25	48
10	trailers	1280x720	23/29	1728

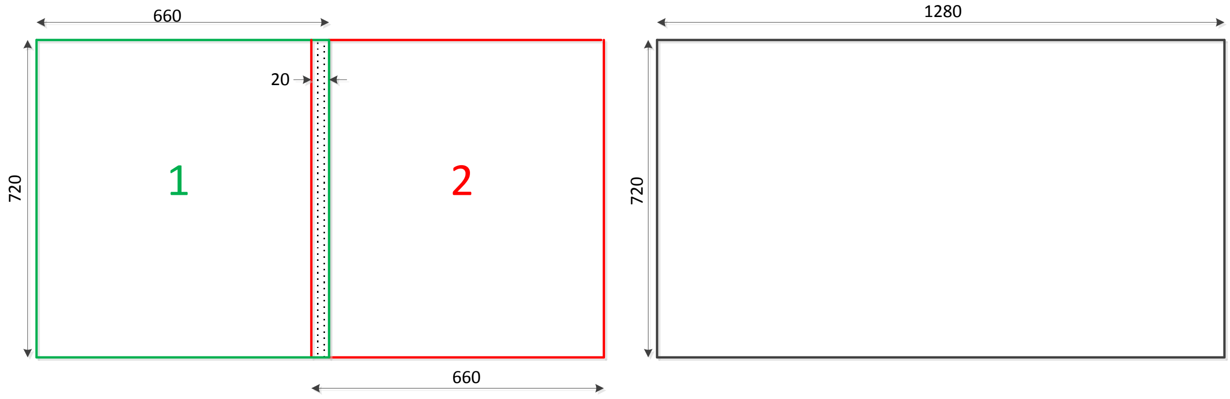
3.2.1 Experiment 1: Estimating the baseline power consumption overhead in tiled setting

We ran an experiment where we played the same video at full resolution (or scale factor k = 1), k=½ resolution (as it would be in a 2-projector tiled display as shown in Figure 3.3) and k=¼ resolution (as shown in Figure 3.2) and measured the power used by averaging the power used across all of the videos in each category. We measured the total power consumption of the system as well as that consumed by the GPU in the same manner. Figures 3.4 and 3.5 show the results respectively. The first thing to note here is that though we expect the bandwidth or power ratios at a scale factor k to be about equal to k (e.g. ¼ and ½ for our 4 and 2 tiled-units respectively), it is much higher in reality. This is due to the overhead of the baseline power consumption, from splitting the video stream and performing the pixel displacement and blending. Thus, we wanted to estimate this baseline power consumption overhead as a baseline. Note that these following equations are derived empirically and validated by the measured data that was shown in Figures 3.4 and 3.5.



(a) k=¼ resolution **(b) k = 1 full resolution**

Figure 3.2 Tiled 4 unit setting – (a) 4 units with k=¼ resolution were measured and compared against to (b) k=1 full resolution – overlapped areas are not shown here



(a) k=½ resolution

(b) k = 1 full resolution

Figure 3.3 Tiled 2 unit setting – (a) 2 units with k=½ resolution were measured and compared against to (b) k=1 full resolution – overlapped areas are shown here

We modeled the power consumed by the i^{th} category at a resolution of scale factor k (k<=1.0) as

$$S + O_{i\_k} + k^{alpha} * P_i = M_{i\_k} \tag{3.8}$$

S: baseline power consumption of the system

O_{i_k} : baseline power consumption overhead at resolution scaling k

$S + O_{i\_k}$: measured total baseline power consumption at idle state

P_i: average power consumption to play the i^{th} category videos

M_{i_k}: measured power consumption

$alpha$: saving variable

Using this equation for three scale factors of k=1, ½ and ¼ and i=1 to 10, we produced a set of 30 linear equations with 11 unknowns ($alpha$ and P_i). We solved the unknowns with the first 11 equations and verify the results on the rest of equation to validate our findings.

$$S + O_{i\_1} + P_i = M_{i\_1} \tag{3.9}$$

$$S + O_{i_k} + k^{alpha} * P_i = M_{i\_k} \tag{3.10}$$

Solving (2), (3), we have:

$$P_i = M_{i\_1} - O_{i\_1} - S \tag{3.11}$$

$$alpha = \frac{\ln(M_{i\_k} - O_{i\_k} - S) - \ln(P_i)}{\ln(\mathrm{k})} \tag{3.12}$$

The calculated $alpha$ is about 0.52 and the power consumption for the scaling system are 0.70 and 0.49 with k=½ and k=¼ respectively. The values which provided all of the GPU power consumption estimations, are consistent with the measured GPU power consumption data, and thus can become a generalized formula to bring about the GPU power consumptions for all of the resolutions in different video categories.

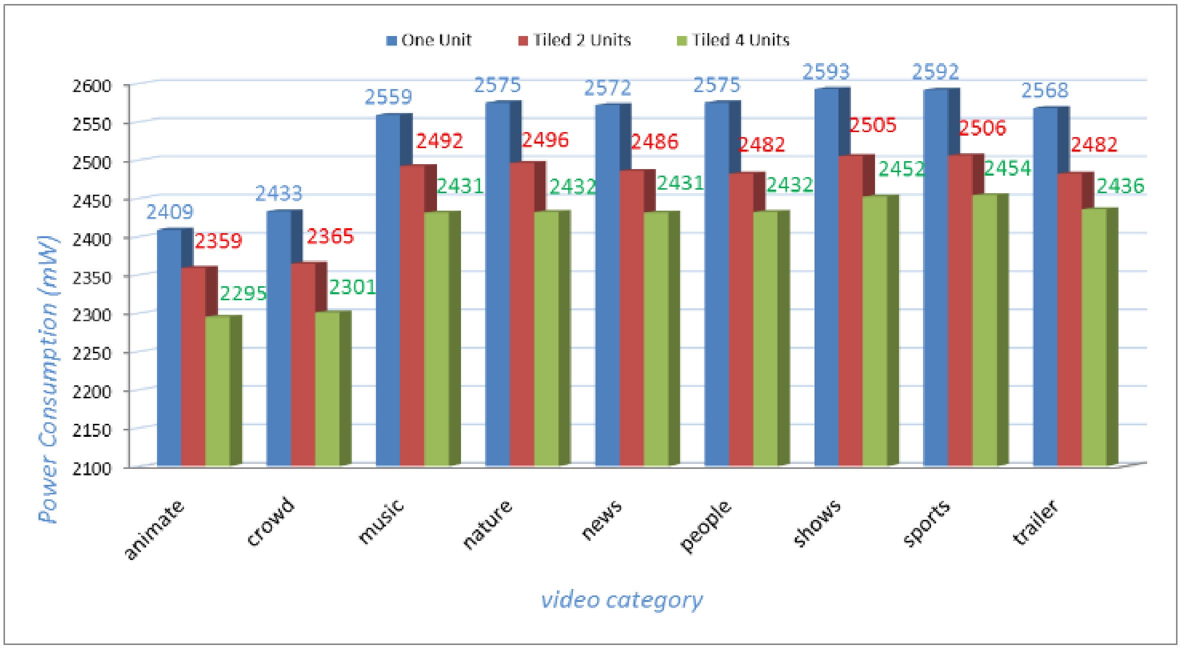


Figure 3.4 System Power Consumption - Tiled Resolution Aggregation

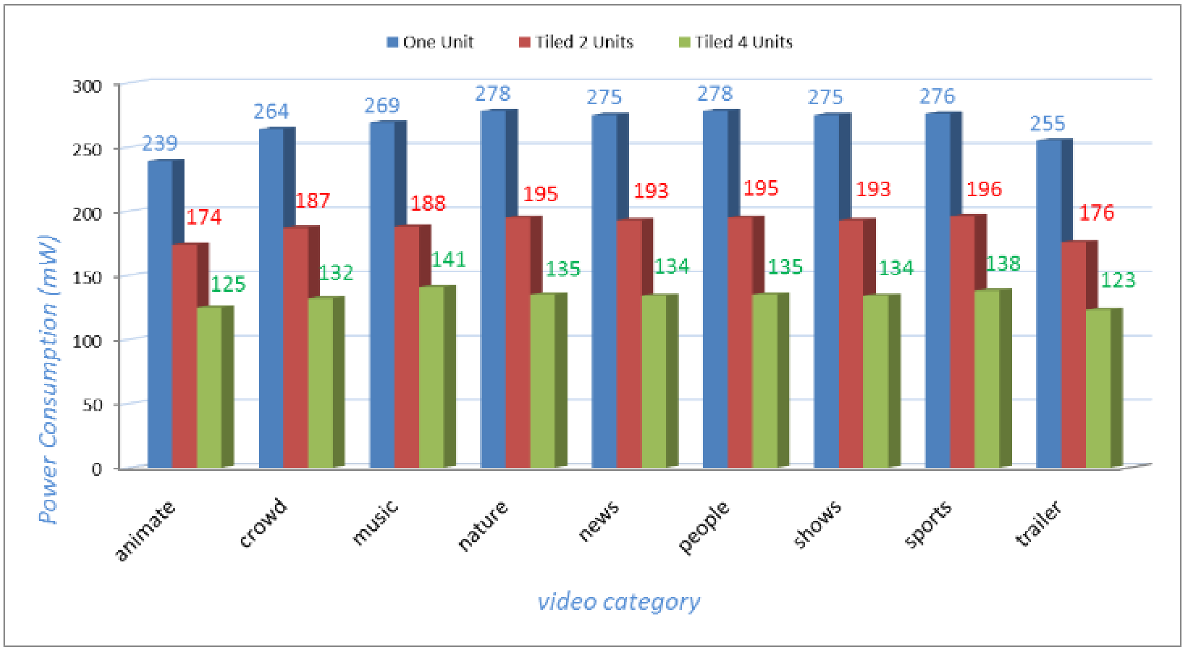


Figure 3.5 GPU Power Consumption - Tiled Resolution Aggregation

Similarly for the bandwidth usage saving, we ran an experiment where we played the same video at full resolution, ½ resolution, and ¼ resolution and measured the bandwidth consumed by averaging the bandwidth consumed across all of the videos in each category. Figure 3.6 shows the result. We modeled the bandwidth usage by the i^{th} category at a resolution of factor k (k<=1.0) as

$$k^{gamma} * B_i = N_{i\_k} \tag{3.13}$$

B_i: average bandwidth consumed to play the i^{th} category videos

N_{i_k}: measured bandwidth consumption at resolution scaling k

Using this equation for three scale factors of k=1, ½ and ¼ and i=1 to 10, we produced a set of 30 linear equations with 11 unknowns ($gamma$ and B_i). We solved this equation to find $gamma$ then the bandwidth usage of different resolutions.

$$B_i = N_{i\_1} \tag{3.14}$$

$$gamma = \frac{\ln(N_{i\_k}) - \ln(N_{i\_1})}{\ln(\mathrm{k})} \tag{3.15}$$

Note that these above equations are derived empirically and validated by the measured data that was shown in the following Figure 3.6.

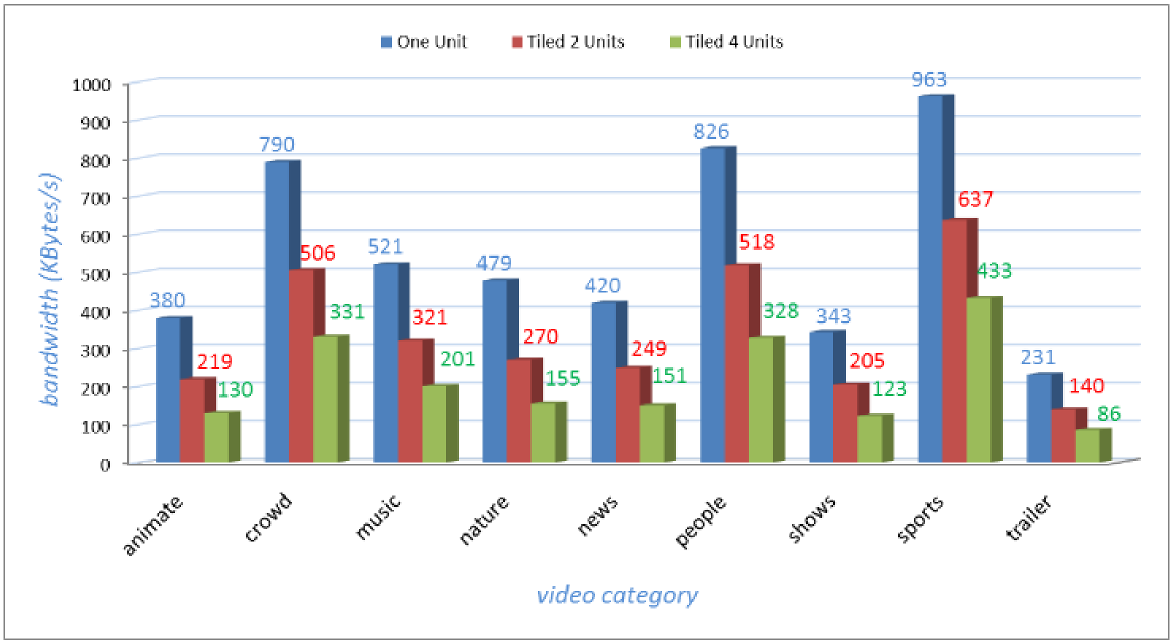


Figure 3.6 Bandwidth Usage - Tiled Resolution Aggregation

3.2.2 Experiment 2: Resource Usage vs. Quality Tradeoff in tiled setting

Next, we wanted to explore how the power and bandwidth usage gets affected with tiling when considering (a) each unit separately; and (b) the federation as an aggregate. Note that we only considered the power consumed by the device for playing the video and not the power consumed by the display. For this, we considered a video resolution of 640x480. First, we measured the power consumed for playing this video on a single mobile device. Note that in this case, we are playing the video on the high resolution mobile display, 1280x960 at a native resolution of 640x480. However, this resolution is on a very small sized display and therefore the perceived resolution is lower since the human eye with 20/20 vision, having a one arcminute resolution gap, cannot detect a pixel density beyond 60 pixels/inch with a typical distance about 5-7 feet from projector to screen, Ref. [18]. Following this, we used a tiled setup of 2 or 4 projectors and requested the scaled version of the video to be played on the federation of projectors. This time we projected the same resolution imagery on a larger space thereby increasing the size and the perceived resolution of the display. However, each projector only achieved a video which was ½ or ¼ the size for 2 and 4 tiled projectors, respectively. This allowed them to reduce their individual power consumption and bandwidth usage though the aggregate went up. Tables 3.2 and 3.3 show the average and standard deviation of power consumption and bandwidth ratios for 2 and 4 tiled projectors. The first column (%) is the percentage of power consumption of individual units in the aggregation setting comparing to 100% power consumption of un-split video on each individual unit. The second column (stdev) is the standard deviation to show the variation of power consumption of different video contents in the same category group. The third column (AGG %) is the percentage of power consumption of the sum of all units in the aggregation setting. Similarly, for columns 4, 5 and 6, we see the percentage of bandwidth usage of each unit, the standard deviation of bandwidth usage among samples in the group, and the total percentage of all units in the aggregation setting, compared to 100% bandwidth usage of un-split video playing on each individual unit.

Table 3.2 Power Consumption and Bandwidth Usage Ratios in Different Video Categories – Tiled 4 Units

Cate-gory	**Power Consumption**			**Bandwidth Usage**		
	%	**stdev**	**AGG %**	**%**	**stdev**	**AGG %**
static	37	3.55	150	33	5.91	133
trailers	45	1.04	179	37	2.08	148
nature	48	1.77	194	34	11.47*	137
people	48	1.61	194	39	2.13	156
shows	49	1.09	195	36	3.35	143
news	49	1.53	195	35	3.63	140
crowds	50	1.66	199	41	9.40*	166
sports	50	1.46	200	44	6.08	176
music	49	0.90	195	38	5.79	152
animate	53	2.42	211	32	9.69*	128
Average	49	2.76	196	37	5.96	150

Table 3.3 Power Consumption and Bandwidth Usage Ratios in Different Video Categories – Tiled 2 Units

Cate-gory	**Power Consumption**			**Bandwidth Usage**		
	%	**stdev**	**AGG %**	**%**	**stdev**	**AGG %**
static	69	1.13	139	57	0.83	114
trailers	73	0.88	146	58	1.04	117
nature	70	2.44	140	57	1.40	115
people	73	1.79	145	58	1.05	116
shows	73	0.81	146	59	1.37	118
news	73	1.59	145	57	1.75	115
crowds	72	1.67	144	57	2.51	113
sports	72	2.56	144	57	1.43	113
music	72	1.15	145	57	3.94	115
animate	72	0.71	143	57	1.82	115
Average	72	1.51	144	58	1.81	115

These tables show that we can increase the size and the perceived resolution in two folds by using a federation of two projectors where each projector consumes on average about 28% reduction of power usage and about 42% reduction in bandwidth usage per user. This indeed is good news for users. But note that this means that the aggregate power and bandwidth consumption increases on average by 44% and 15%, respectively. Also note that when we consider the 4-tiled display, the savings per projector is greater (51% for power and 63% for bandwidth) but the aggregate goes up more (to 96% and 50%). However, the quality of experience has increased by almost 4 times in terms of size of the display and perceived resolution. The point to be aware of here is 4 times the quality is achieved at a lower resource per projector and at a lower than 4 times aggregate resource. Specifically, for our setting, there was about 3% and 4% overlap in horizontal and vertical directions. Therefore, the viewing experience improvement was around 3.85 times instead of 4 times. Also, note that as the number of projectors in the federation increases, the reduction of resource usage for each projector is higher but this comes at the cost of some increases in the aggregate power consumption.

Now, let us take a more careful look at exact numbers for videos in different categories in 4-way tiling. The power consumption of slow motion categories (e.g. static) is about 37%, for medium motion categories (e.g. trailers, nature, people) about 45-48%, and for higher motion categories (e.g. crowds, sports and animation) going up to 50-53%. Although the motion levels are mixed in each video category group due to different video contents so that the high motion group are not differentiated clearly to the slower motion group, we still can observe the relationship between the motion of video content and the power consumption saving. We conclude that the savings ratios are inversely proportional to the amount of motion in the video. This can be intuitive due to the sequences of large motion vectors that have less inter and intra frame compression than those with smaller motion vectors.

Additionally, note that while the standard deviation of power consumption seems to be small in all categories, the standard deviation of bandwidth is noticeably higher for nature (11.47), crowds (9.40), and animation (9.69). We observe that this is due to the high variation of motion strength which is a derivative of the amount of motion. This leads to the conclusion that bandwidth ratios have a dependency on the variation of motion strength.

We have not observed any impact on the playing with these mismatched ratios due to the variations which are offset by the input video buffers. Consequently, each unit still maintains their synchronization during the entire playing time with our prototype. However, research on the effects of these mismatched bandwidth ratios according to their motion strength may be an interesting subject wherever the input video buffer resource is limited and the resolution is very high.

3.2.3 Low power consumption with aggregated high resolution in tiled setting

Figure 3.7 shows the projection of the power savings for a full range of resolutions. Consider that in lower resolution videos, there is not much of a gain in using a federation. But with higher resolutions, this relation is super linear. Shown in Figure 3.7 (the orange branch of the integrated four units) is one example which demonstrates where nearly UHD 4K 3840x2160 is achieved by a very low-cost system with GPU power consumption of only

300mW for each unit that plays at 1080p. This is very promising for the capability of projector enabled mobile devices to deliver very high definition and quality video in the future. This tiled setting provides a flavor of how resources can be managed across federations to maximize viewing experience in term of resolution.

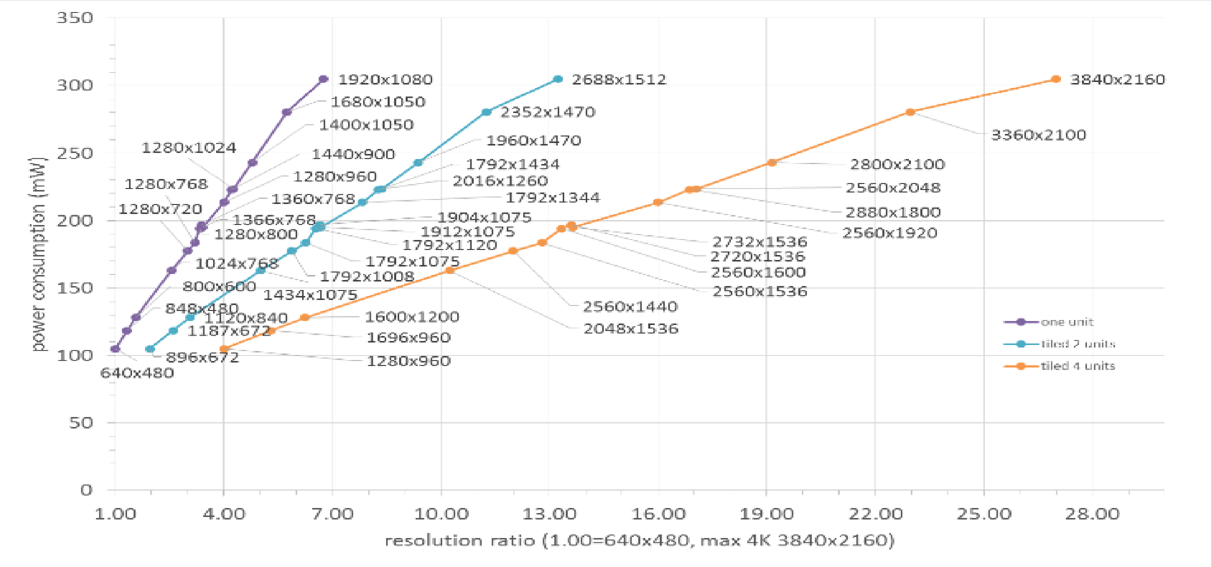


Figure 3.7 Low Power Consumptions in Resource AggregatedSystems with Higher Resolutions

3.2.4 Experiment 3: Estimate the baseline power consumption overhead in a superimposed setting

In the next two experiments, s***uperimposed setting experiments,*** we superimposed two or four projectors and measured the power consumption and bandwidth usage for each projector set. The goal here is to reduce the resource usage per projector by reducing the frame rate of each projector by a factor of total number of projectors used in the federation.

For example, for a 2 projector superimposed display, each projector launches every other frame alternatively. Therefore, each of their frame rates reduces by a factor of two, but the frame rate of the federation remains the same.

For this, we took each video and ran it at different frame rates using a single projector, 2 superimposed projectors and 4 superimposed projectors then measured the power usage as shown in the graphs below. Let the power required to project each frame be P_f, and the baseline power consumption of the system be S, and the baseline power consumption overhead be O. Therefore, we model the power consumption as:

$$S + O + f * P_f = M_f \tag{3.16}$$

where f is the framerate and M_f is the measurement of the power usage in each frame rate. Note that the above equation is derived empirically and validated by the measured data that shown in Figures 3.9 and 3.10.

We use measurements presented on the graph in Figure 3.8 to solve a linear regression and find the P_f, S and O. Note that S is a baseline power consumption of the system, and not dependent on the frame rate variable (i.e. 1707mW given by the black horizontal line in Figure 3.9) and O is a significant value, and therefore is the predominant resource that is used at lower framerates (i.e. observing the small change of power consumption at the low frame rate, 5fps on the right column in Figure 3.10, compared to their baseline overhead value)

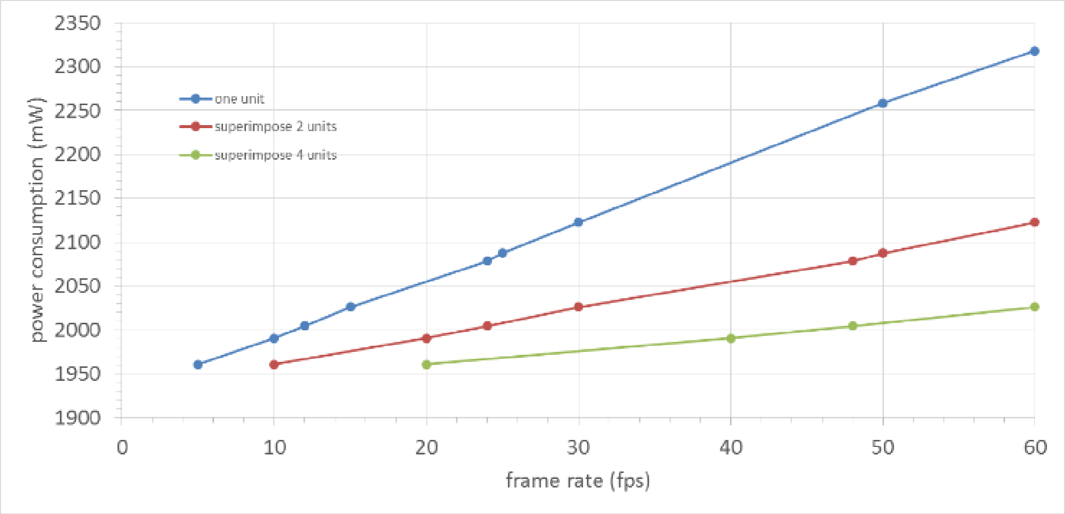


Figure 3.8 Superimposed Frame Rate Aggregation Graph

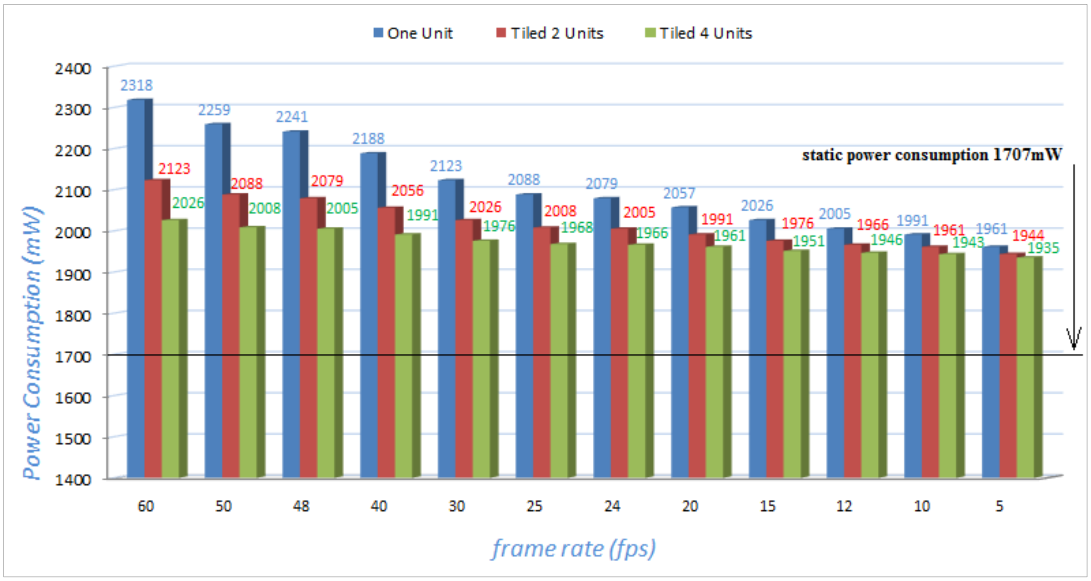


Figure 3.9 Total Power Consumption – Superimposed Frame Rate Aggregation

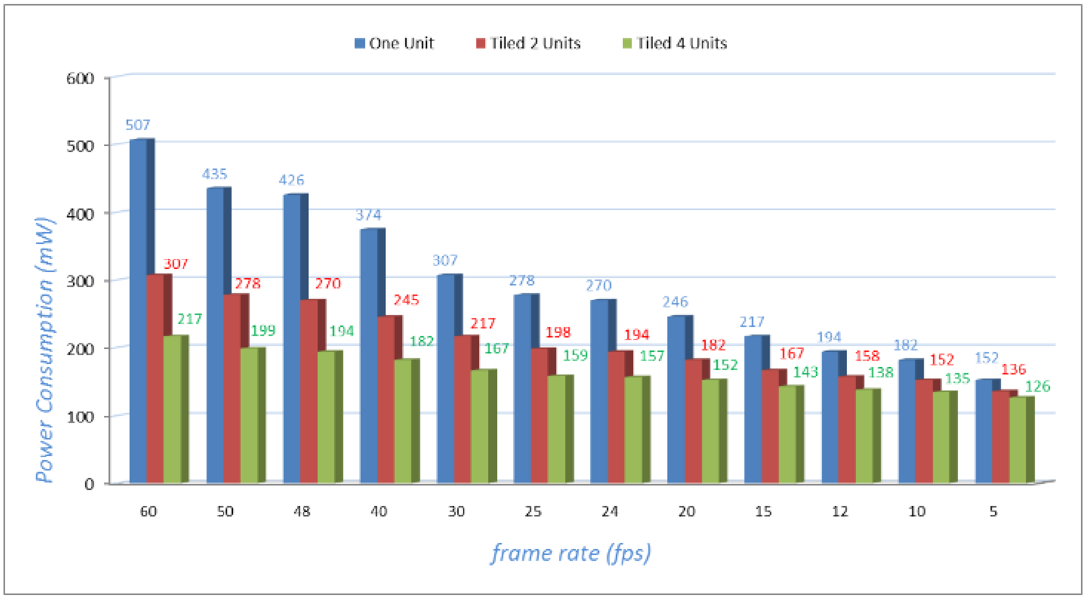


Figure 3.10 GPU Power Consumption – Superimposed Frame Rate Aggregation

However, when we do the same measurements for bandwidth, we found that bandwidth usage (as shown in Figure 3.11) did not change much as the frame rate reduced by significant factors in a federation. We relate this to the fact that during the transcoding of the video and transmission, it was difficult to skip frames due to the way the video encoding is performed using P, I and B frames. To that end, the dropping of frames only affects the power that is consumed due to their display.

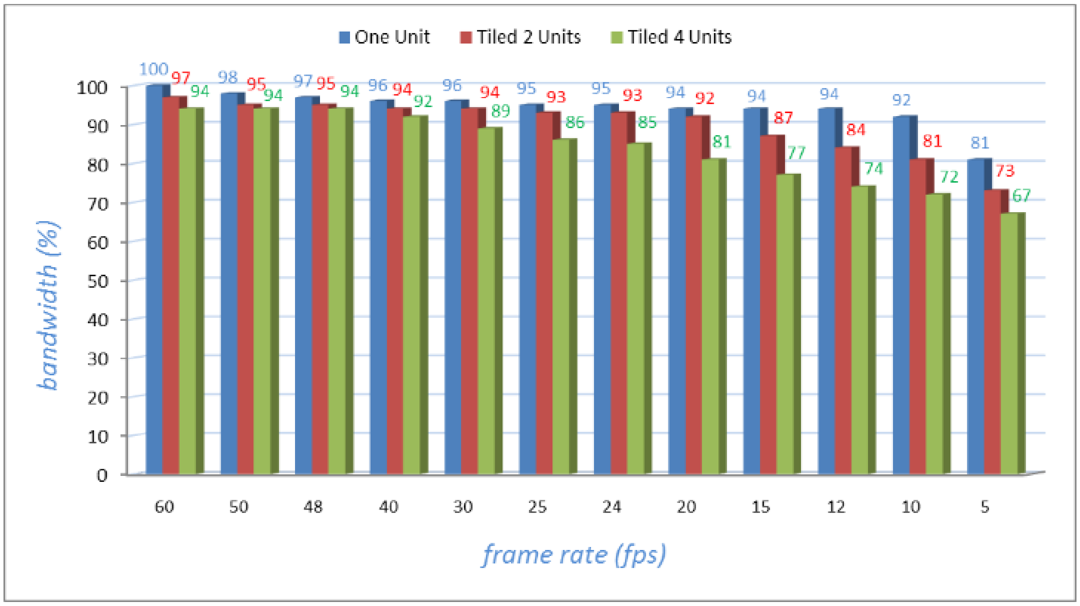


Figure 3.11 Bandwidth Usage – Superimposed Frame Rate Aggregation

3.2.5 Experiment 4: Resource Usage vs. Quality Tradeoff in a Superimposed Setting

Table 3.4 shows the power and bandwidth consumed for each unit when two projectors were superimposed. The first column (%) is the percentage of power consumption of individual units in the superimposed aggregation setting compared to 100% power consumption of un-split video on each individual unit. The second column (stdev) is the standard deviation to show the variation of power consumption of different video contents in the same category group. The third column (AGG %) is the percentage of power consumption of the sum of all units in the aggregation setting. Similarly, columns 4, 5 and 6 show percentages of bandwidth usage of each unit, standard deviation of bandwidth usage among samples in the group, and the total percentage of all units in the aggregation setting, compared to 100% bandwidth usage of un-split video playing for each individual unit.

Table 3.4 Power Consumption and Bandwidth Usage Ratios in Different Video Categories – Superimposed 2 Units

Cate-gory	**Power Consumption**			**Bandwidth Usage**		
	%	**stdev**	**AGG %**	**%**	**stdev**	**AGG %**
static	74	0.69	147	99	0.10	198
trailers	77	1.22	155	91	5.40	183
nature	74	3.15	148	96	1.60	192
people	78	1.62	157	95	0.81	189
shows	76	1.48	152	97	1.03	193
news	78	1.71	156	96	1.05	191
crowds	77	1.92	155	94	4.44	188
sports	78	1.65	156	90	0.32	198
music	76	2.00	152	95	8.04	190
animate	76	3.09	152	87	1.08	174
Average	77	1.98	153	93	2.64	187

Note that each device reduced the power usage by more than 27% of power, while the aggregate went up. However, since there is not much bandwidth savings per unit due to the superimposed aggregation, if each unit has enough power to run the videos, it is better to superimpose them at full frame rate to achieve twice the brightness. As discussed earlier, the portability and power efficiency features of pico-projectors come at the cost of a severely reduced image quality in terms of brightness and resolution. This superimposed setting provides a flavor of how resources can be managed across federations to maximize viewing experience in term of brightness.

3.3 Measurement Details

Figure 3.12 shows the system setting for the power consumption measurement. The current clamp probe Tektronix TC305 is used with the amplifier AC/DC Tektronix TCPA300 which output is connected to and calibrated with the oscilloscope Tektronix DPO3054. We set the oscilloscope to its maximum 5M points with a 10 second length screen. This setting has the resolution 500K samples per second which is high enough to collect all of the dynamic variations of power consumption during playing. Then we used the area calculation in the entire measured period to derive the total power consumption. The unit under test is the prototype system with three components in each projector embedded mobile device as described in Chapter 2, section 2.2.

A proposed practical way to verify the reliability of our measurement is to look into the resolution of the measured data and the deviation of the outcome. Our results are very accurate with different video content loads, less than 1%, and also very consistent for every re-measuring in less than 0.1% difference among various re-measures. Moreover, in order to compare the un-split vs split video on the same video content frames, we verified the differences between each re-measure during playing time and stop playing time, in a consistent reference sampling time as follows: (a) wait 10 seconds after playing starts to prevent the unstable power consumption during video loading period; (b) take an integration of 10 sampling seconds for playing time; (c) wait 10 seconds after playing stops to ensure the processor had already finished doing all of the closing video processes; (d) then take integration of 10 sampling seconds for measuring the power consumption during idle time. The re-measurement variations at those sampling points are just a few in several thousand units which is very consistent to less than 0.1% error and can be taken for every measurement to ensure the accuracy of each measurement.

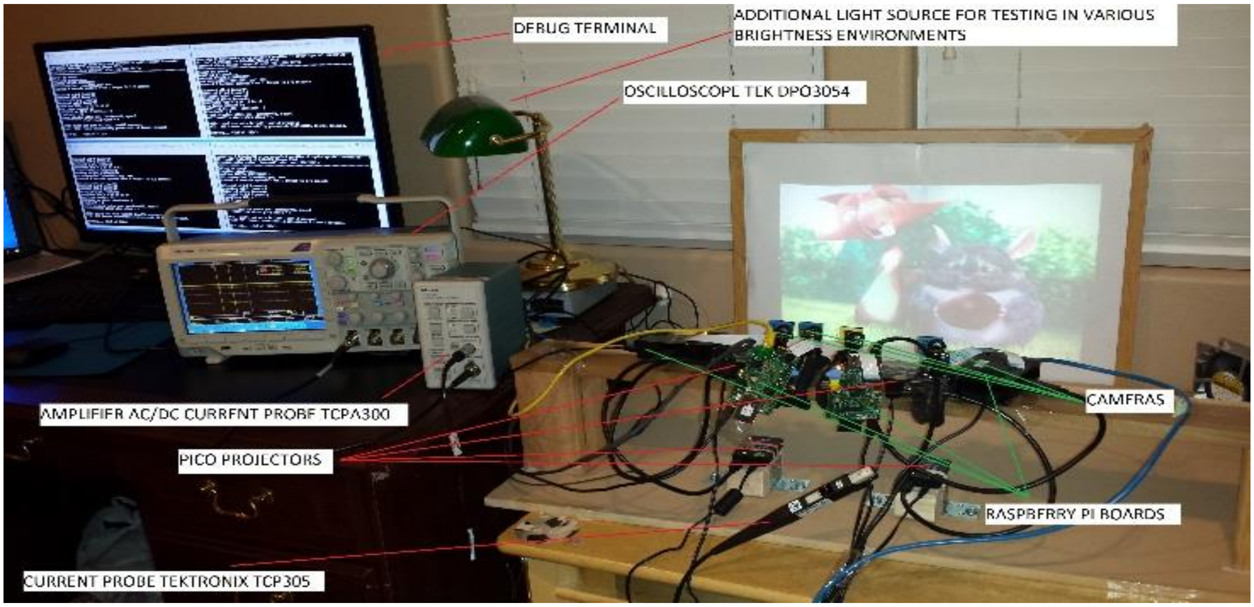

DEBUG TERMINAL
ADDITIONAL LIGHT SOURCE FOR TESTING IN VARIOUS BRIGHTNESS ENVIRONMENTS
OSCILLOSCOPE TEK DPO3054
AMPLIFIER AC/DC CURRENT PROBE TCPA300
PICO PROJECTORS
CAMERAS
RASPBERRY PI BOARDS
CURRENT PROBE TEKTRONIX TCP305

Figure 3.12 Projector Enabled Mobile Device with Measure Equipment

3.4 Dynamic Voltage Frequency Scaling in the Federation System

The next issue we investigate is if there is a possibility of reducing the aggregate power consumption further thereby making this technique even more amenable to practical use, especially from the perspective of service providers. In order to achieve this, we proposed the application of a *Dynamic Voltage Frequency Scaling (DVFS)* technique on the aggregation system to reduce power consumption further by lowering the supply voltage and operating frequency.

Nevertheless, there is a trade-off here between the longer battery life due to the reduced power consumption via DVFS and system performance. The battery life savings is not significant for higher video resolution since it requires higher pixel frequency as well as processing frequency. Nonetheless, as our collaborative video in the tiled setting specifically reduced the resolution of the video to be played by each device, we anticipated that we can optimize power consumption further with lower voltages and frequencies via DVFS and yet still maintain acceptable quality of services.

DVFS reduces the switching losses of the system by selectively reducing frequency F and voltage V of the system as per a well-known power consumption relation equation:

$$Power \sim \tfrac{1}{2}CV^2F \tag{3.17}$$

Applying it in our system, we took advantage of this in the tiled setting with lower resolution videos to downscale the voltage to further save power. Ideally we could achieve the voltage downscaling settings in our system from 1.2V to 1V, a 31% dynamic switching power savings in equation (3.17).

On our prototype system with all of its peripheral components, and with camera operating, we measured the power consumption by changing core voltage from 1.2V to 1V. This voltage scaling was achieved without compromise to the video quality and system performance. This resulted in around a 5% savings as shown in Figure 3.13. Since our prototype model came with external and peripheral components that have not been applied with voltage downscaling, the voltage scaling on the core of SoC alone can only produce an extra 5% reduction in power consumption compared to the total system power consumption. In any case, in a practical setting with SDRAM that can be downscaled in mobile device platforms and when the measurement is on SoC alone, this extra power savings from voltage downscaling values should be higher. For example, we re-measured on our prototype loading with our custom built kernel to disable all of peripherals i.e. LAN, the USB Hub, and the camera, and then reevaluated the power savings with voltage downscaling and got a 14% reduction in power consumption instead of 5%.

The measured power consumption savings of the system was about 8% from 1.2V to 1.0V or projected to doubling it, about 16%, with scaling from 1.2V to 0.8V which was flowed from the core's dynamically switching power savings. Moreover, in terms of the frequency, the reductions for core and SDRAM frequencies also yielded a measured total power consumption (static and dynamic) of 14%. The combined total of extra power savings of the two voltage and frequency scaling settings (about 22%) were measured on this collaborative system without compromising the video quality and system performance. Therefore, the combined extra power consumption savings, about 22% by measurement, up to 30% by projected estimation, using the voltage and frequency downscaling method, was suitable for our federation setting. Such method can provide extra savings of power consumption since the resolution of playing video on each device in the federation is lower than the un-split video resolution on the single device.

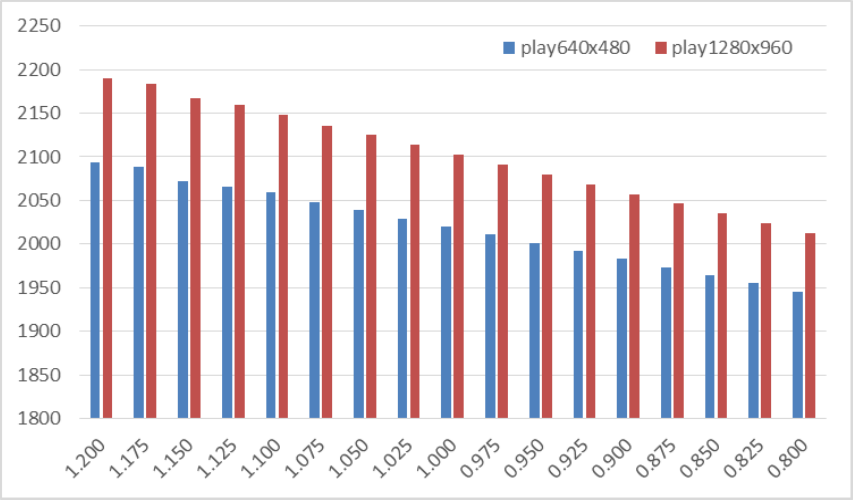


Figure 3.13 Power Consumption (mW) vs. Core Voltage Downscaling (V)

3.5 Conclusion

We proposed a low-cost, low power, embedded system which uses a visual feedback from a camera to perform the registration, video synchronization, and video transformation processes in real time to display partitions of video by multiple pico projectors. This aggregate resource, in terms of a power and bandwidth embedded system is a comprehensive federation solution model for projection-enable mobile devices. These tiled and superimposed collaborative settings aim to bridge the gap between the video quality expectation and deliverability of the projection-enabled mobile devices in the future. By providing a real time solution for the collaborative federation system with higher performance and significant resource savings in terms of power consumption and bandwidth, the proposed system has shown that it can be used as a model for the federation of mobile devices to enhance the user's viewing experiences.

In conclusion, the projector enabled mobile device with low-cost, low power consumption design, through aggregated resource settings, either tiled or superimposed, is able to achieve very high definition and higher brightness. It shows that the federation solution can be used for the projector enabled mobile devices in the future to enhance the video viewing experience for users and also to improve the resource savings substantially.

Chapter 4

Auto Re-Calibration

4.1 System Overview

Our proof-of-concept setup consists of multiple tiled or superimposed pico-projectors each connected to an embedded system with a camera to capture the projected video as similar in Chapter 2. The block diagram for the methods in each of these prototype systems is shown in Figure 4.1 for both a tiled setting and a superimposed setting. Note the path C in green for the differences of the auto re-calibration task. We provided the same additional capability of (a) ability to achieve video streaming applications that demanded real-time performance; and (b) ability to perform at lowest power budget by exploiting GPU based compression techniques. Moreover, the auto re-calibration requires another method of registration and image transformation that can perform concurrently with playing video process during the presentation. The Raspberry Pi board version 2 was used to substitute the version B in the system described in Chapter 2 to get better performance for this task.

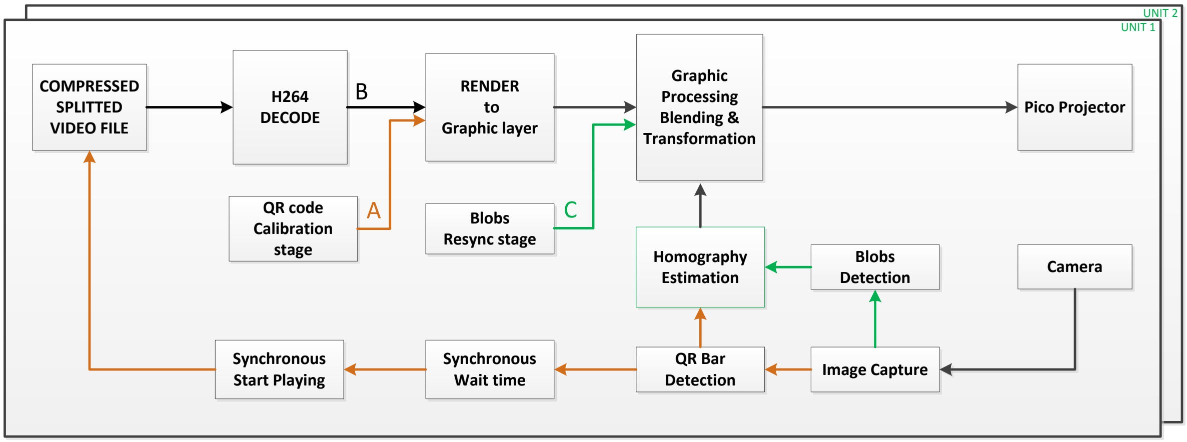


Figure 4.1 Block Diagram of Auto-Resynchronized Tiled Setting – path A: calibration stage, path B: video presentation stage and path C: resynchronizing stage

4.2 Auto Re-Calibration Overview

As discussed earlier, portable, low-power, light weight, small size pico-projectors are key components of projection-enabled mobile devices for the future. Due to the reduction of weight and dimension and the portability nature of the projector-enabled mobile devices, the calibrated integrated systems are prone to physical un-stabilizing of the projected image during the presentation. Thus the auto re-calibration during the presentation time becomes an essential requirement to enhance user experience.

However, from the calibration stage and presentation stage separately algorithm as discussed in Chapter 2 to simultaneously run both re-calibration and presentation are several major issues:

- Issue 1: the visual feedback methods with QR cannot be used due to the presentation video.
- Issue 2: the hardware speed limitation to run the video capture thread and video playing thread simultaneously.

For dealing with issue 1, we evaluated two types of calibration patterns and used dot patterns instead of QR barcodes. Note that based on visually feedback methods investigation in mobile device:

- QR bar code: has ability to embed information, is suitable for adding frame number information and coordinate parameters but takes a long time to detect especially in the noisy environment.
- Dot: does not have ability to embed extra information in the code. Advantage: simple, fast detection, is suitable for feedback and coordinate adjustment in real time.

In order to alleviate the impact of issue 2 on video performance, frame rate and capable presentation resolution, we chose a higher power SoC, Rasberry Pi 2, and also performed auto re-calibration for each and every second instead of every frame. It can take a few seconds to do the re-calibration but due this short time latency to recover, did not impact user experience.

4.3 Auto Re-Calibration Solution

From Figure 4.2, let p1 is the coordinate of projected points, s1 is the coordinate of display image of these points and c1 is the coordinate of captured image of these projected points. Simple relationships among them are:

$$p_1 \rightarrow s_1: \qquad s_1 = h_{ps1} * p_1 \tag{4.1}$$

$$s_1 \rightarrow c_1: \qquad c_1 = h_{sc1} * s_1 \tag{4.2}$$

$$p_1 \rightarrow c_1: \qquad c_1 = h_{pc1} * p_1 \tag{4.3}$$

$$p_2 \rightarrow p_3: \qquad s_2 = h_{ps2} * p_3 \tag{4.4}$$

Since $p_3 = H_c * p_2$, (4) can be rewrite as:

$$s_2 = h_{ps2} * H_c * p_2 \tag{4.5}$$

Also, in order to have s_2 stitching to s_1,

$$s_2 = h_{ps1} * p_2 \tag{4.6}$$

From equations (4.5) and (4.6), the compensation homography can be calculated as:

$$H_c = h_{ps2}^{-1} * h_{ps1} \tag{4.7}$$

Since $h_{pc1} = h_{sc1} * h_{ps1}$ =>

$$h_{ps1} = h_{sc1}^{-1} * h_{pc1} \tag{4.8}$$

and $h_{pc2} = h_{sc2} * h_{ps2}$ =>

$$h_{ps2}^{-1} = h_{pc2}^{-1} * h_{sc2} \tag{4.9}$$

Substitute equations (4.8) and (4.9) into (4.7), we have $H_c = h_{pc2}^{-1} * h_{sc2} * h_{sc1}^{-1} * h_{pc1}$ and since $h_{sc2} = h_{sc1}$, the derived compensation homography will be:

$$H_c = h_{pc2}^{-1} * h_{pc1} \tag{4.10}$$

Thus H_c is derived into that closed form can be determined from h_{pc1} and h_{pc2}, *the* two known homography matrices. These matrices were determined from the coordinate of the known dot pattern coordinates, p_1 and p_2, and the detected coordinate of the captured images of these patterns on the camera, c_1 and c_2.

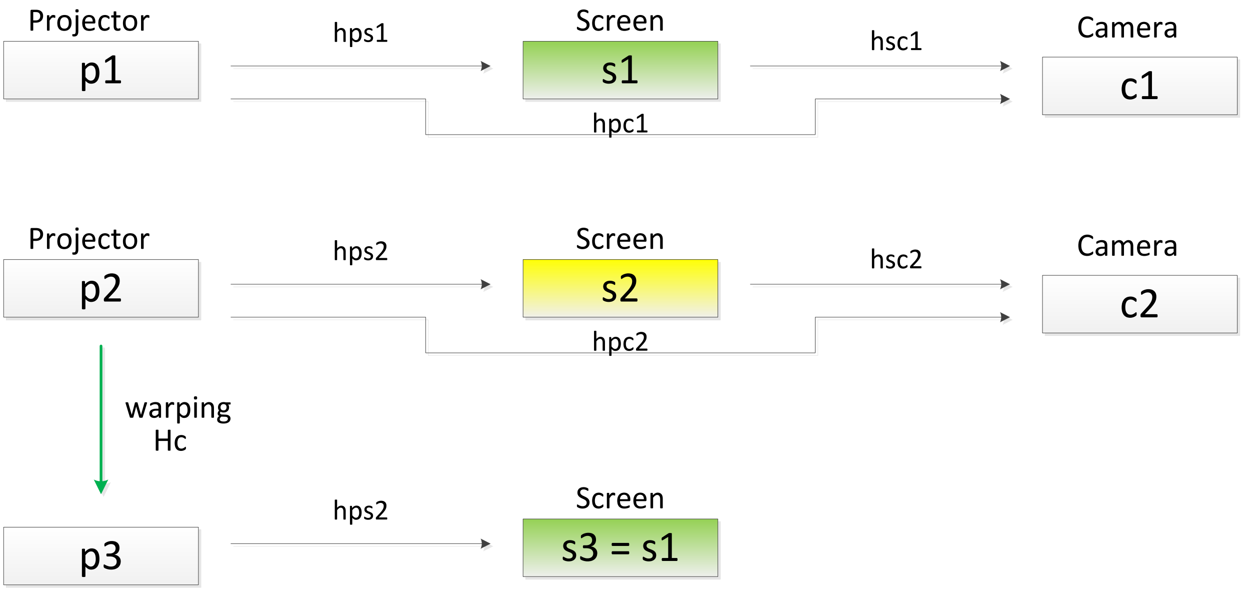


Figure 4.2 Two Units Stitching Together by Warping Video Image after Calibration Stage

Figure 4.3 shows the equivalent system of the two-projector model after stitching. After the units were calibrated, it continued to check and adjust during playing to keep this equivalent state.

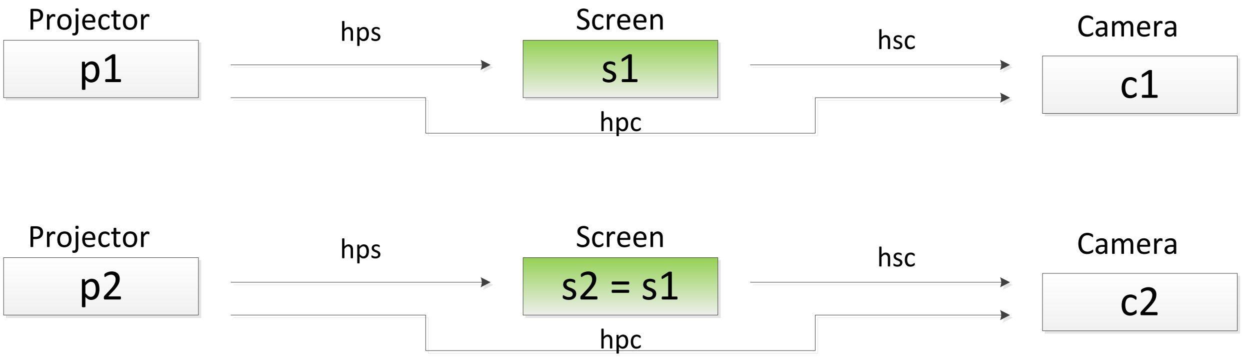


Figure 4.3 Equivalent Model for Two Units after Stitching Together by Warping Video Image

When projector 1 is moved, the projected image on the screen will be changed from s_1 to s_1'. However, due to camera and projector appended on the rigid body as in the projector enabled mobile device model, the captured image of the patterns on the camera were still the same, c_1 . Since projector 2 is not moved, the projected image on screen remains the same, s_2 . The captured image of screen s_2 on camera 1 now changed to c_2'.

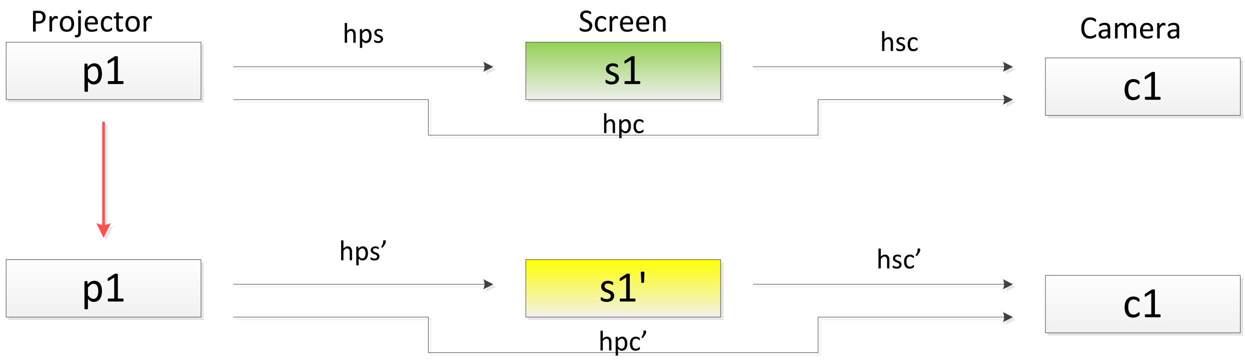


Figure 4.4 Unit 1 with Projector and Camera in a Rigid Body Moved to a New Location

In order to correct the projected screen image from s_1' back to the original projected screen, s_1 , we introduce the adjusting homography H_a as shown in Figure 4.5 so that:

$p_1 \rightarrow p_a$:
$$p_a = H_a * p_1 \tag{4.11}$$

Since warped image, p_a , will be projected to the recovered original positions on screen,

$p_a \rightarrow s_1$:
$$s_1 = h'_{ps} * p_a \tag{4.12}$$

$$s_1 = h'_{ps} * H_a * p_1 \tag{4.13}$$

Compare to (4.1), we have

$$H_a = h'^{-1}_{ps} * h_{ps} \tag{4.14}$$

We translate the right side of equation to some values that we can calculate to find H_a as follows:

$$h_{pc} = h'_{sc} * \ h'_{ps} \tag{4.15}$$

$$h'_{ps} = \ h'^{-1}_{sc} * \ h_{pc} \tag{4.16}$$

$$h'^{-1}_{ps} = \ h^{-1}_{pc} * \ h'_{sc} \tag{4.17}$$

Substitute (4.17) into (4.14), we have:

$$H_a = \ h^{-1}_{pc} * \ h'_{sc} * \ h_{ps} \tag{4.18}$$

Or finally,

$$H_a = \ h^{-1}_1 * \ h_2 \tag{4.19}$$

With $h_1 = h_{pc}$ that can be calculated from the pre-defined pattern p_1 and the detection of captured pattern image c_1 and

With $h_2 = \ h'_{sc} * \ h_{ps}$ that can be calculated from the information of the pre-defined pattern p_2 and the new captured image c'_2 as follows:

$p_2 \ \rightarrow \ s_2$: $$s_2 = h_{ps} * \ p_2 \tag{4.20}$$

$s_2 \ \rightarrow \ c'_2$: $$c'_2 = h'_{sc} * \ s_2 \tag{4.21}$$

Thus $$c'_2 = h'_{sc} * \ h_{ps} \ * \ p_2$$

Or $$c'_2 = h_2 * \ p_2 \tag{4.22}$$

In this equation (4.22), p_2 is the known coordinates of the patterns to project, c'_2 is the coordinates of the new captured of the patterns. Thus from these known point values, we derived the homography h_2 then finally the adjusting homography H_a in equation (4.19). Table 4.1 shows the accordingly algorithm to find the adjusting homography during playing time.

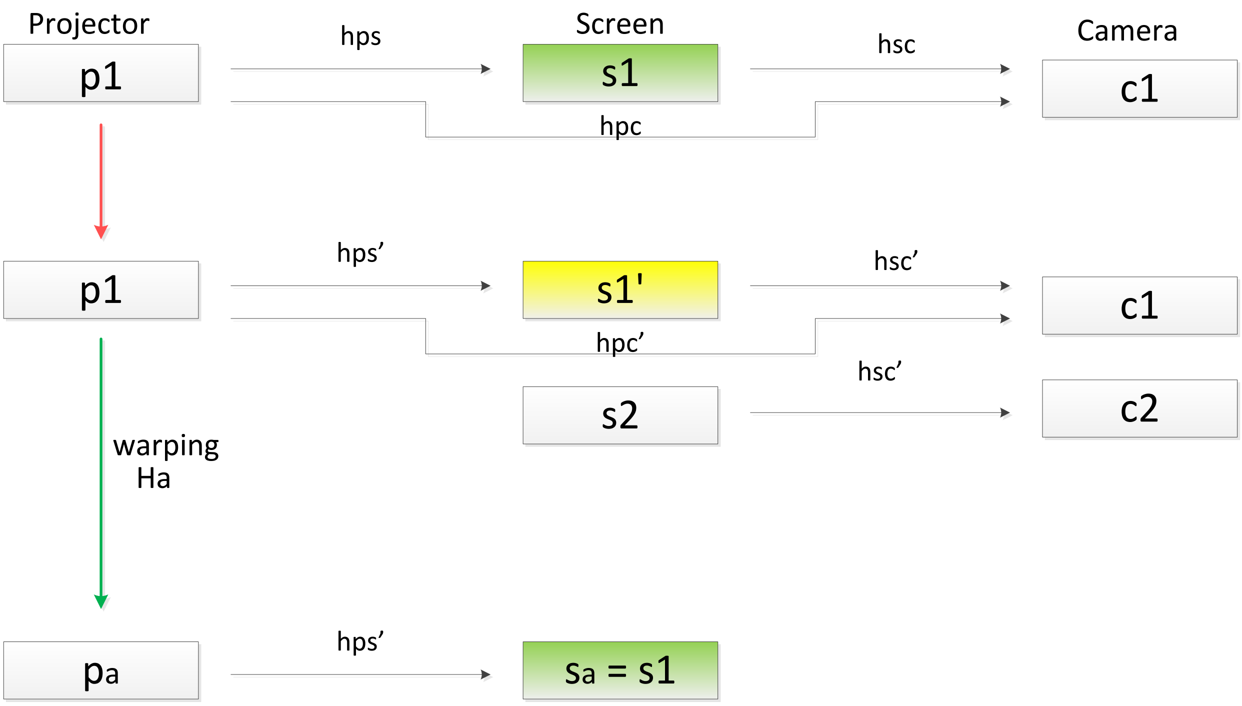


Figure 4.5 Adjustment of Screen Back to Stitching Position by Warping Video Image

Table 4.1 Auto-Calibration Algorithm*:*

1. First stage calibration to find H_c
2. Warp video images and play
3. Project dot patterns with the warped coordinates, p_1 and p_2
4. Capture the projected dot patterns on screen s_1, normal operation, c_1
5. Calculate homography h_1 then h_1^{-1} from (4.3)
6. Capture the projected dot patterns on screen s_2 , after projector was moved, c_2'
7. Calculate homography h_2 from (4.21)
8. Calculate homography H_a
9. Apply the adjusting homography H_a to warp the video images
10. Go back to step 6 to continue running the auto adjusting loop

4.4 Implementation and Result

4.4.1 Initialization Stage

Figure 4.9 shows the implemented Software Flow Chart of auto recalibration process in which the resynchronization stage is performed simultaneously with video presentation stage. In the initialization stage, first the CPU has been set up with the video streaming path or load the movie locally depending on the input source selection. Also, the CPU loads the QR codes to be used for camera-based registration from its media storage SD card and projects them sequentially. Simultaneously, the CPU captures the whole projected image with its embedded camera. That was shown as green path A in the Figure 4.1. The frame data, which was embedded in QR codes, is extracted from each captured image. The CPU also graphically detects the corners of the QR codes and the centers of the blobs embedded in them from the captured images. It continues the capture and extract processes until the desired data can be collected, validated, and used. This desired data contains two sets of information: the set of coordinates for features embedded in the QR codes and the set of frame number parameters. The frame number parameter has been used in this stage to synchronize each projector with its frame number delay accordingly. The coordinate parameters will be used in the next step, the following video presentation stage.

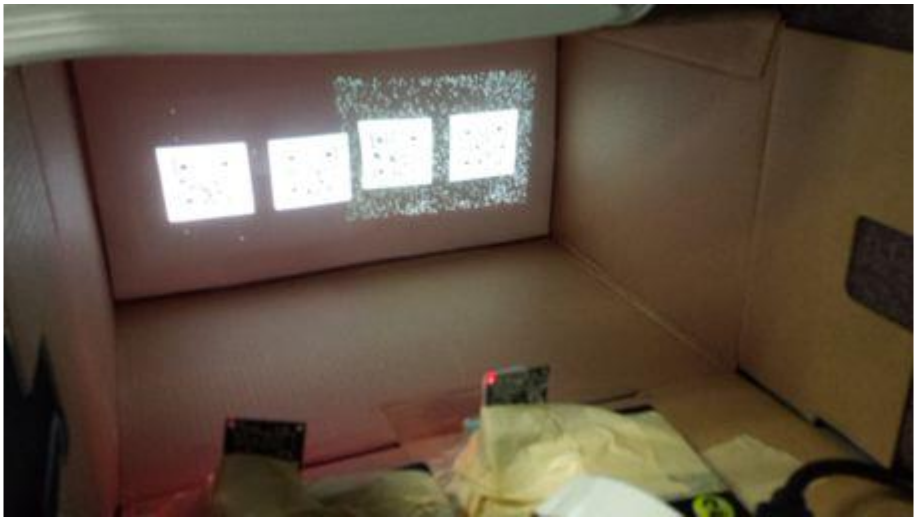
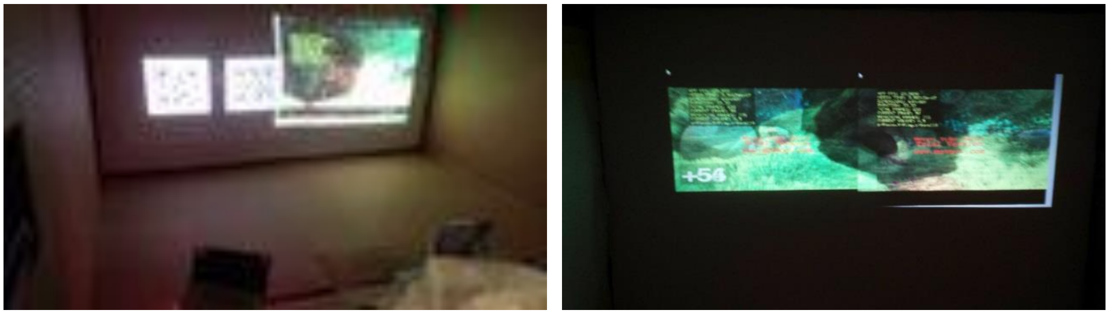


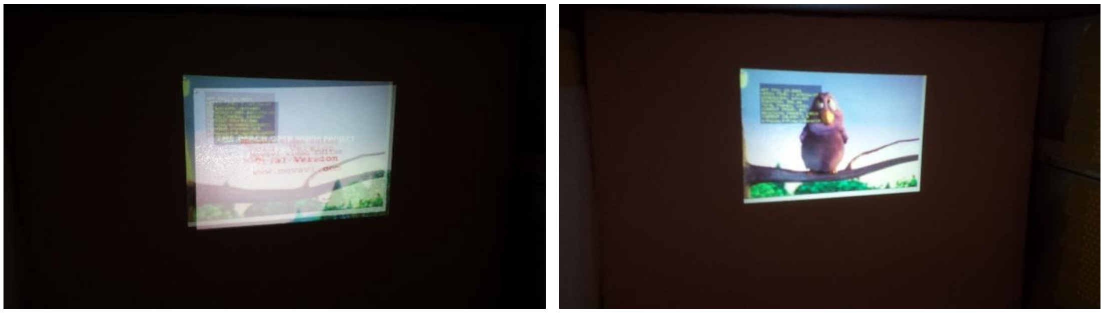
Figure 4.6 Two Tiled Re-Synchronous System Setting



(a) Before re-calibration

(b) After re-calibration

Figure 4.7 Two Tiled Re-Synchronous System



(a) Before recalibration (b) After recalibration

Figure 4.8 Two Superimposed Re-Synchronous System

4.4.2 Video Presentation Stage

The video presentation stage as shown in the Figure 4.9 is actually a pipeline loop in which each loop is needed to be done in one frame period. The blending process in which pixels are manipulated partially for the interested area is performed by CPU. The video transformation process which performs the warping to entire image is performed by GPU openGL. The sharing resource processing plays an important role to ensure video presentation maintain in real-time with high frame rate.

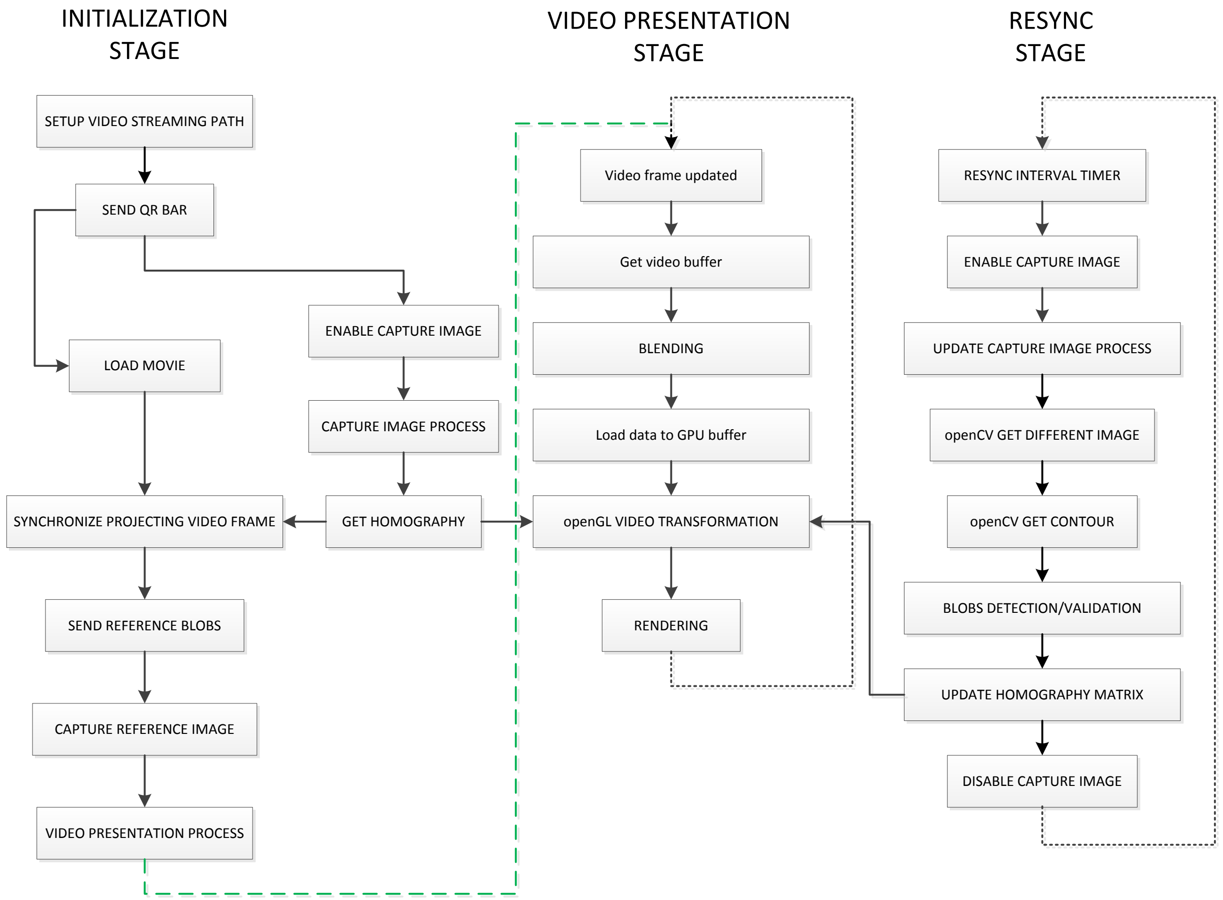


Figure 4.9 Software Flow Chart of auto recalibration process – resync stage is performed simultaneously with video presentation stage

4.4.3 Re-synchronization Stage

In the re-synchronization by using a visual feedback from a camera to perform the auto-registration during the presentation time, a fast pattern is generated outside of the projected video to detect and correct the displacement of the projector. To minimize the impact on user experience, an interval timer is set to check the alignment regularly but not all the time. This method can be substituted with an IMU assisted method to detect the movement of projectors then use the visual feedback to do the correction. This maybe a subject for a future research. When the interval timer is expired, the camera capture process is enabled to capture the dot patterns. According to the differences with previous dot pattern coordinates, it will re-update the homography matrix and send to video transformation in video process to do the correction. Then it turns off the camera capture process until the next cycle of the updating interval timer. All of the detection and correction processes have been taken place in about one second. For the aggregated system, this much latency does not impact on the user experience since the displacement of projectors is happened occasionally. Meanwhile, seamlessly feature is a critical requirement for a video stabilization system when the movements of projectors are essential. This subject of video stabilization will be discussed in the next chapter.

4.4.4 Results

We developed the software on two MPPP units as described in previous section 4.1 with Raspbery Pi model 2. After registration and synchronization stages, to perform the auto-registration during the presentation time, we generated a dot pattern outside of the projected video when the timer expired and capture it back. Since the camera capture works simultaneous with decode process, we notice a bit delay for each frame processing but still meet the frame rate 30 fps with Raspberry Pi model 2 hardware. The H264 Decode engine still completed its decoding task less than 30ms for each frame. This stage meets the timing requirement to be less than the frame period which is 33ms. In parallel, there is capture process as shown in Figure 4.9, RESYNC column. In this process, the “UPDATE CAPTURE

IMAGE" process takes 16ms, the openCV "GET DIFFERENT IMAGE" process takes about 16ms, then the "GET CONTOUR" process takes about 23ms. Thus this path takes about 2 frames latency to update to the correction homography to adjust the video image back to its aligned position. One limitation in our implementation is the range of displacement cannot be large that the video content can overlap to the outside dots. This overlapping can interfere the dot coordinate detection outcomes. To increase the range of displacement, one can do two step adjustments, the use of IMU assisted method can be used to do the coarse adjustment then applying the visual feedback to perform the fine adjustment.

4.5 Conclusion

We proposed a novel method which uses a visual feedback from a camera to perform the auto-registration during the presentation time. Since this feature is performed during the presentation time, the specific dot patterns have been used and the real time processes have to be arbitrated so it does not affect the video presentation. The auto correction can be done in a second thus not seamlessly. In the next chapter, we will present a similar model with enhanced hardware that can perform the correction almost seamlessly with this visual feedback method. Alternately, the use of both IMU assisted method to detect the movement of projectors and visual feedback method to do the correction can improve the detection response time while still maintain the accuracy of image alignment. This maybe an interesting subject for a future research.

Chapter 5

Projected Video Stabilization

5.1 Introduction

Portable, low-power, light weight, small size pico-projectors are key components of projection-enabled mobile devices for the future. Due to the reduction of weight and dimension and the portability nature of the projector-enabled mobile devices, stabilizing the projected image during the presentation becomes an essential requirement to enhance user experience. In this Chapter, we proposed a design and two implementation methods to address this stability issue. The first method uses a camera-based visualized feedback where an artifact marker is generated from the projector and/or a fixed marker set on the screen is used. For descriptive ease in this chapter, we named this method "Camera-based Assisted Projector Stabilization" and will refer to it as 'CAPS'. With CAPS, the camera captures the current image and compares it to the previous image to sense the movement of the projector. Then geometry parameters are derived to calculate the compensation matrix. This compensation matrix is then used to warp the video image to give the same projected image on the screen. This CAPS method compensates accurately over long time periods but works well only in slower motions. For faster motion projector stabilization, we proposed the second method with the Inertial Measurement Unit (IMU) assistance to estimate movement and predict the compensation parameters. We named this method as "IMU Assisted Projector Stabilization" (IAPS). This IAPS method responded well in faster motion and proved to be able to compensate in all rotation movements. The IMU used in the prototype is a low-cost, generic, common module with a combined accelerometer, a gyroscope, and a magnetometer to assist motion tracking for stabilizing compensation.

5.2 Related Work

5.2.1 CAPS Method Related Work

Camera-based visualized feedback technology has been used popularly in the visualization community, mainly for auto calibration of standard projectors. Its technology is based on projection of an artificial pattern with visualized pattern feedback recognition to perform the registration, Ref. [5]. There is extensive literature of camera calibration techniques in the robotics and photogrammetry field which is listed in Ref. [47]. These techniques digitize one or more pictures of an object of fixed and sometimes unknown geometry, locate features on the object, then use mathematical optimizers to solve for the viewing parameters. In such applications, geometric representative homography and an eight-point algorithm were mentioned in Ref. [19][20] and [21] respectively. The above applications are focused into the calibration stage for multiple standard projectors. Raskar *et al*. in Ref. [24] presented a fast registration technique that can be auto calibrated in less than 15 seconds which focused on the system's calibration process. That author also used warping technique to correct images when projectors are in a rough position, Ref. [23]. Sukthankar *et al*., Ref. [22], exploited the homography in camera-projector systems. However, the calibration was performed only once during the calibration stage and no lagging time was of concern in these studies.

In this Chapter, for the CAPS method, in addition to using these two mentioned techniques dealing with standard projectors for registration, we propose an auto-registration technique that can correct the projected video all the time and has very small latency to have a nearly unnoticeable effect for stabilizing. The latency requirement is also a motivation to explore a simple homography calculation algorithm, a simple color dot pattern, and implement it on a low-cost embedded device in a way to utilize both GPU and CPU resources.

5.2.2. IAPS Method Related Work

For this IAPS method, instead of using camera visualized feedback as in the CAPS, an Inertial Measurement Unit (IMU) is attached onto the projector enabled mobile device to track the orientation. The IMU has been used popularly in different application aspects. In the visualization field, Raskar *et al.*, Ref. [26], presents an IMU assisted method that helps to calibrate the image distortion in a few seconds. In human motion tracking, various studies using IMU have been performed to track limb segment orientations relative to each other or to calculate joint angles, as opposed to estimating the orientation of a limb segment relative to an Earth-fixed reference frame. These studies focused on finding different estimation algorithms that suit their specific application requirements. Two popular types of estimators, complimentary filter and Kalman filter, have been developed and implemented in many studies.

Yan-De *et al.,* Ref. [27], used inertial and magnetic sensors and estimated the fusion data based on a complementary filter. Yun *et al.,* Ref. [30], and Bachmann *et al.*, Ref. [33], presented a motion tracking system that is based on the MARG (Magnitude, Angular Rate, and Gravity) sensors which included inertial and magnetic sensors. Calusdian *et al.,* Ref. [28], improved the complementary filter with adaptive-gain method for orientation estimation. Madgwick in Ref. [29] proposed the Gradient Descent algorithm. Note that in these complementary filter systems, the gyro bias drifts causing orientation error growing over time and are eliminated by fusing with accelerometer and magnetometer. The system errors and sensor noises have not been accounted for by these filters and thus the accuracy depends on the sensor. Then the Kalman filter, suitable for estimating the system errors among different types of data, was implemented and optimized in various studies. Wang *et al.,* Ref. [32], designed a quaternion-based Kalman filter that fuses angular velocity with observation quaternions produced by Adaptive Step Gradient Descent. Marins *et al.*, Ref. [31], improved the mentioned Bachmann *et al.,* Ref. [33], by substituting the complementary filter with a Kalman filter. Foxlin in Ref. [35] described the design of a Kalman filter in the head tracker application. Sabatini in Ref. [36] presented a quaternion-based Extended Kalman Filter (EKF) with 9 DOF sensors and then evaluated its performance in Ref. [37].

This chapter uses a similar approach to design an approach for fusing the IMU data to use in the IAPS method. The proposed approach was designed to meet the following specifications: The first specification is very low latency where the latter has to match the video refresh rate (i.e. a few tens of milli-seconds instead of a few hundred milli-seconds compared to other applications). In order to meet this requirement one has to consider the latency of sensor data collection, the filter response, and the warping video processing time. The second specification has more stringent angular accuracy. The angular data to sense the orientation in the IMU is translated to a linear pixel displacement in the projected screen. Due to any small angular data error can result in a noticeable shift on the screen, that angular data accuracy requirement is essential in our system. However, in a commercial projector enabled mobile device, the IMU typically is a common low-cost sensor. Thus, the fusion filter has to be a key to compensate for the IMU's shortcomings and deliver the accuracy, responsiveness, and eliminate the noise from the sensing data.

5.3 Main Contribution

This chapter presents two novel approaches to mobile projector stabilization. The first approach is the Camera-based Assisted Projector Stabilization (CAPS) and the second approach is the IMU Assisted Projector Stabilization (IAPS).

5.3.1 CAPS Method

Unlike standard projectors, the calibration in projector enabled mobile devices has to be performed during the video playback. This leads to the latency in correction which becomes a major issue. In this chapter, we present an approach to solve the stabilization or auto-calibration in real time. Given the low brightness of mobile pico-projectors, the patterns have to be designed and calibrated to screen out the background noise while making sure the projected pattern does not interfere with the playback video content. We proposed a new method to exploit the image-based feedback system in which the camera captures the projected images in real time and auto adjusts the projected screen position automatically.

Our contributions on this CAPS method are:

(a) Architect a projector-enabled mobile device with an embedded camera to stabilize a projected video in real time

(b) Implement this visualized feedback position with simplified dot pattern referencing and updating method. The main proposal of this method is to cut off the video processing time which requires major computerized resource and time.

(c) Develop an algorithm to perform auto-visualized compensation.

5.3.2 IAPS Method

In the IAPS method, we designed and implemented an IMU assisted system with a modified version of EKF predictor. Our contributions on this approach are:

(a) Architect a projector-enabled mobile device with attached IMU to stabilize a projected video in real time

(b) Implement the IAPS with the IMU fusing design using a simplified version of an EKF algorithm and optimize the warping video process to run faster in order to meet the more stringent timing requirement of the projector stabilization

(c) Demonstrate the stabilized projector enabled mobile device prototype *with* and *without* compensation as a proof of concept system

The rest of this chapter is organized in the following way - Section 5.4 describes System Models and Prototypes. Section 5.5 describes the Camera-based Assisted Projector Stabilization method (CAPS). Section 5.6 presents the IMU Assisted Projector Stabilization method (IAPS). Section 5.7 presents rotation compensation solutions. Lastly, Section 5.8 presents the conclusion and closing remarks.

5.4 System Models and Prototypes

CAPS and IAPS consist of two main tasks, SENSE and DISPLAY. The SENSE block tracks the motion and send it to the DISPLAY block to warp the video output. In the CAPS system, the SENSE block has the projector launch the reference dots while the camera captures then detects these reference dots to track the movement of the projected image. The output of the SENSE block in CAPS is the compensation homography that it then sends to the DISPLAY block. Based on the estimated homography, the DISPLAY block performs video warping in real time to do the compensation. More detail will be presented in section 5.5. In the IAPS, the SENSE block contains the IMU with 3-axis gyroscope, 3-axis accelerometer, and 3-axis magnetometer sensors. Further detail can be found in section 5.6.

In order to build a proof of concept system, we considered two major approaches. The first approach is using an existing mobile device and developing software on that platform. The advantages of this approach are simpler, have a quick start, and cut-off development time in both software and hardware. This is a software-based approach and is well suited for software applications and algorithm improvements for the current hardware platform in the market. Another advantage of this approach is the quick deliverable from concept to application.

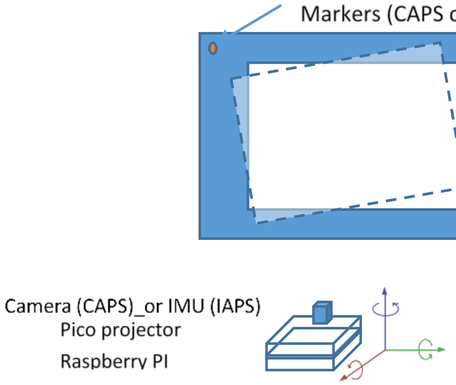


Figure 5.1 Simple Block Diagram of CAPS/IAPS Systems

The second approach is architecting a prototype system, selecting a hardware platform and interfacing different hardware devices in that system. We chose this hardware system approach to give more flexibility in design because of the following reasons. First, the latency requirement in projected video stabilizing is a key to meet and validate. Mobile devices are overloaded with software tasks and are unlikely to meet those requirements. Second, working with video processing on a low-cost embedded system is different from on desktop or server platforms since resources are very limited, requiring more low level access to balance the resource allocations. Third, the stabilization application requires more hardware flexibility in order to select and be able to add peripheral devices.

Our prototype is targeted to emulate relevant parts of a projector-enabled mobile device. Next, we describe our platform, again namely, a Mobile Plug and Play Projector (MPPP) with modified components from the system from Chapter 2.

This MPPP now has three components:

(a) The first component is a low-cost but newer version, version 3, Raspberry Pi 3 board Ref. [13] that contains the Broadcom SoC BCM2837, a high definition 1080p Embedded Multimedia Applications Processor. This SoC contains a quad ARM core Cortex-A53 Applications Processor (CPU) and a dual core VideoCore IV® Multimedia Co-Processor (GPU) with 1080p30 Full HD High Profile H.264 Video Encode/Decode engines. Again, this component is still a typical low- cost, low power, low resource hardware platform that was chosen as a processing unit similar to a typical low-cost mobile device platform. It was also chosen for our prototype due to its popular, open source software accesses for experiments and applications.

(b) The second component is a typical, low profile pico projector suitable to be embedded into a mobile device platform, Ref. [12].

(c) The last component is a typical low-cost camera that has a five mega pixel resolution, which supports up to 1080p 30Hz, 720p 60Hz, or VGA resolution with a refresh rate up to 90Hz

For CAPS, we use this low-cost three-component MPPP unit as a projector-enabled mobile device model. We built a camera-based prototype system in a loose physical setting. This system performs self-adjusting for stabilizing the video during playing with sensing and measuring of coordination parameters through visualized feedback from the camera. The compensation task is performed by the GPU on the display video stream.

For IAPS, the camera is substituted by a 16 bit resolution IMU that has 9 DOF (Degree of Freedom). The IMU combines 4 sensors with typical specifications, a gyroscope with full range +/-2000dps with 70mdps/LSB sensitivity, an accelerometer with full range +/-8g with 0.25mg/LSB, and a magnetometer with full range +/-4 gausses with 0.16mgauss/LSB, Ref. [58][59].

5.5 Camera-based Assisted Projector Stabilization (CAPS)

5.5.1 Estimation Compensation Matrix

For ease in following, we listed two well-known equations (5.1) and (5.2) in computer graphic, Ref. [17], then derived the stabilizing equation for CAPS. The homography matrix can be estimated from the relation of each projector's coordinates to the coordinates of the observing camera. Note that a homography implicitly assumes a flat or planar display surface.

At the initial phase or anytime a user wants to get the reference, the reference homography can be captured by the image displayed on the camera. This image is broadcasted to the MPPPs and actually contains blobs that are outside of the playing video on the fly. These blobs are alternatively flashing. The captured images reconstruct a contour of the blobs and search for the central.

Let us consider N feature points in the dot patterns of the projector *i*. Let the known coordinates of the i^{th} feature point, $1 \leq i \ \leq N$ in a projector's coordinate system to be *(x_i, y_i)*. Let the corresponding point detected in the camera's coordinates by (X_i, Y_i).

These two corresponding points are related by this well-known equation:

$$\begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \tag{5.1}$$

Converting Equation (5.1) to linear equations, we get:

$$\begin{pmatrix} -x_i & -y_i & -1 & 0 & 0 & 0 & x_iX_i & y_iX_i & X_i \\ 0 & 0 & 0 & -x_i & -y_i & -1 & x_iY_i & y_iY_i & Y_i \end{pmatrix} (H)^T = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{5.2}$$

when all the N feature points are written out in this form, we get the matrix $AH^T = O$ where A is a 2Nx9 matrix, H = (h1, h2, h3,...h9) and O is a 9x1 null vector. When considering a planar surface, the homography matrix has 8 degrees of freedom, and hence h_9 is assumed to be 1. Therefore, we have to estimate the eight unknowns of H from this equation. Note that if we have a large number of features, this leads to an over-constrained system. Therefore, we find the values of H using a linear least square minimization. Generally this process of finding the homography is repeated for each time to find the homography i.e. h_{sc1} from the relation of point from screen to camera $s1 \rightarrow c1$ that represented in the Figure 5.2.

Thus at initial stage, we get h_{ref1} from set $p_1 \rightarrow c_1$ which p_1 is the coordinates of the 4-projected reference blobs and c_1 is the capture image of those blobs:

$$h_{ref1} = h_{pc1} \tag{5.3}$$

We also get h_1 from set $s_1 \rightarrow c_1$ which s_1 is the coordinate of the markers and c_1 is the capture image of those markers:

$$h_1 = h_{sc1} \tag{5.4}$$

When the device is moving, a new position introduces a new set of reference. We also get those estimation values:

$$h_{ref2} = h_{pc2} \tag{5.5}$$

$$h_2 = h_{sc2} \tag{5.6}$$

Note that $h_{ref2} = h_{ref1} = h_{ref}$ so we don't need to take reference again. h_2 is achieved from the set of markers to the new captured image of those markers.

For the correction, the warped video p_3 that generated from compensation new matrix H_c ($p_3 = H_c * p_1$) has to project the same scene s_3 ($s_3 = h_{ps2} * p_3$) that was displayed in the original scene s_1 or $s_3 = s_1$. Derive this equation to find the compensation matrix H_c as follows:

$$h_{ps2} * p_3 = h_{ps2} * H_c * p_1 = h_{ps1} * p_1 \tag{5.7}$$

$$h_{ps2} * H_c = h_{ps1} \tag{5.8}$$

Substitute h_{ps1} from

$$h_{pc1} = h_{sc1} * h_{ps1} \tag{5.9}$$

and substitute h_{ps2} from

$$h_{pc2} = h_{sc2} * h_{ps2} \tag{5.10}$$

into (5.8), we have:

$$h_{sc2}^{-1} * h_{pc2} * H_c = h_{sc1}^{-1} * h_{pc1} \tag{5.11}$$

$$H_c = h_{pc2}^{-1} * h_{sc2} * h_{sc1}^{-1} * h_{pc1} \tag{5.12}$$

Note that the values on the right hand side of Eq. (5.12) were known from the results of Eq. (5.3)-(5.6), thus the compensation matrix is obtained by:

$$H_c = h_{ref}^{-1} * h_2 * h_1^{-1} * h_{ref} \tag{5.13}$$

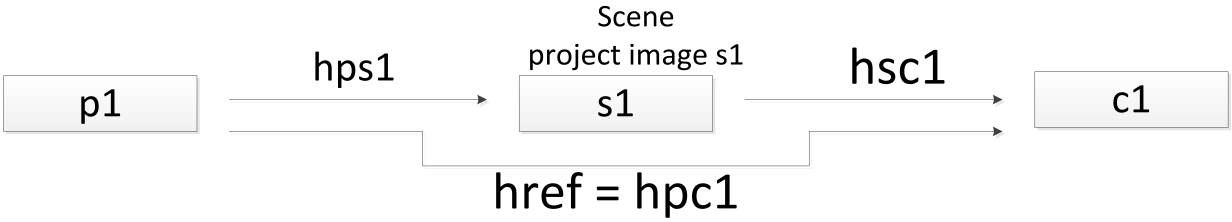


Figure 5.2 Original Reference Homography Matrix

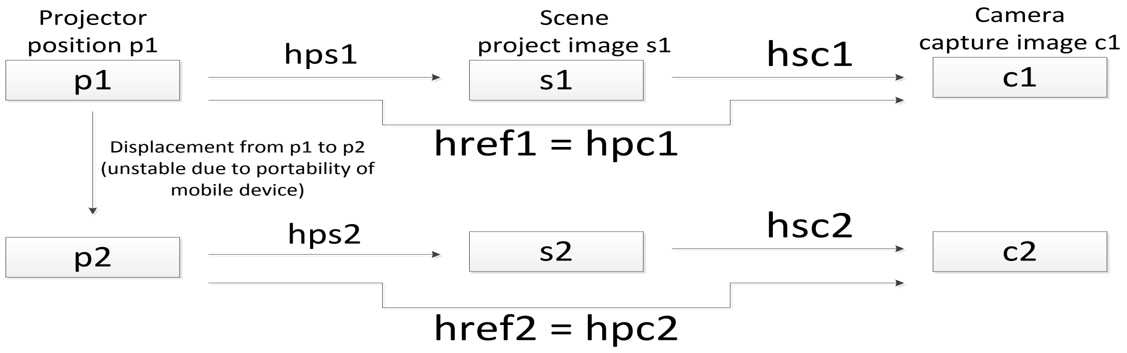


Figure 5.3 Projector Displacement

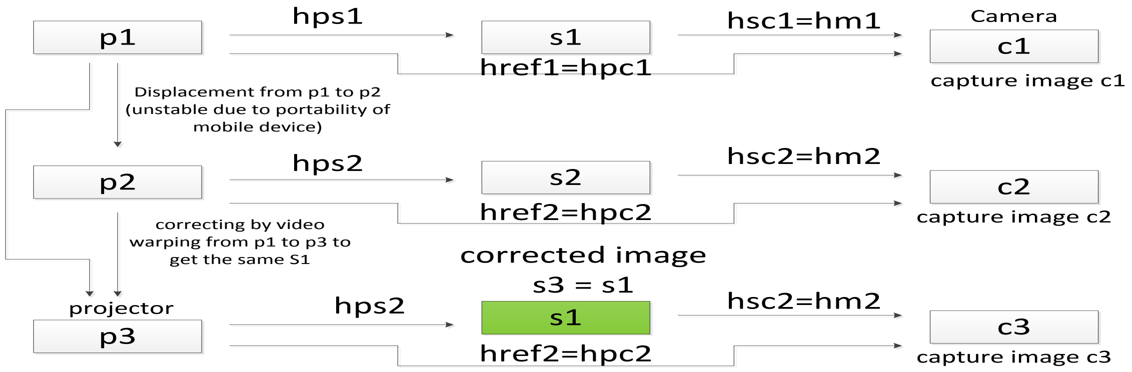


Figure 5.4 Corrected Projected Video

5.5.2 Implementation and Result

In the SENSE block implementation, the challenge in this approach is reducing the large video lagging due to the intensive video processing and capturing. The camera has to capture the updated reference frame simultaneously, detect the pattern, and derive the compensation parameters while the video decoding engine is on duty. We balanced these processing loads among the GPU and CPU in our typical low-to-medium resource and horsepower embedded system to achieve the maximum video resolution of the system.

In the DISPLAY block implementation, every frame needed to be warped using the estimated homography in real time as well. This required decompressing the video content then processing its frames individually. Thus this geometric registration via pixel displacement guided by the homography was another timing challenge for low-cost embedded systems, especially when playing high resolutions, i.e. 1080p, at 30 fps. We exploited the GPU engine to decode the H264 video, then rendered the decoded video to the graphic layer where the pixel manipulation for homography transformation was done in real time. Note that by doing the homography estimation in the one-time initiation stage and the homography transformation seamlessly during play time, we have achieved real time operations of 30fps at high definition resolutions of 1080p.

In the timing consideration, with this approach, we only achieved a nearly seamless video stabilization with transition from the original position to the displacement position and corrected back to the original position. The reason is due to the principal of this approach since it has a “post correction” nature in which the compensation matrix is constructed after the video is displayed to the displaced position. As a result, this method stabilizes well in slow motion but not in fast motion, due to that latency. The detected position does not account for the latest position that leads to “half compensation” status. We noticed some small displacement toward the direction of the movement in fast motion. This is a motivation to explore the second approach with IMU assisted to predict the orientation earlier. We will present this approach in the next section.

5.6 IMU Assisted Projector Stabilization (IAPS)

The presented video stabilization using camera feedback works well to stabilize slow motion and is very stable for a long period of time. For stabilizing faster movement, however, due to the “post correction” nature as mentioned earlier, there is the time latency that affects the correcting position that causes offsets from the original position. Moreover, although fixed passive markers are not needed with a non-equipped projector mobile device connected to a projector in a loose setting, i.e. a mobile device with an external stand-alone pico projector, they are needed with the projector-enabled mobile devices. Once the projector is embedded into the mobile device in a rigid body, the fixed passive marker is required in addition to the active projected markers to supply another frame reference for motion tracking. This marker does cause an inconvenience for the user. Such limitations of the first method - CAPS - motivated us to find another solution in which an IMU is used to sense the movement of the device in real time. The fused sensing data from the IMU will be used to warp the video images to compensate for the movements. As a result, the projected video will be stabilized or locked.

Figure 5.5 shows the block diagram of the IAPS SENSE block. The sensor block contains an IMU and a distance sensor. The IMU's raw data consists of rotational data from a gyro sensor, acceleration data from an accelerometer, and magnetic field data from a magnetometer sensor. In the fusion block, the orientation tracking is accomplished by integrating the angular rate data from a gyroscope to determine orientation. However, due to the integration errors and quantization errors, they are prone to drift over an extended period of time, Ref. [34]. In order to avoid such drift, two additional complementary sensors, a 3-axis accelerometer and a 3-axis magnetometer are used for referencing the gravity and magnetic field vectors, respectively. These references are fused in the first stage (the TRIAD algorithm block), thereupon the outputs are fused with the gyroscope data in the second stage, Complimentary Kalman filter block. These blocks will be discussed in more detail in the following paragraph.

Our main focus for the fusion block design is an effective filter in which our first priority is real time response. Then our next priority is the accuracy specification in the short term, and finally we addressed accuracy in long term operation. Note that while most of the studies in human body tracking applications and navigation applications implement filters to address accuracy in a long term operation as a primary consideration, our projected images have to be compensated in real time, making the latency issue our primary requirement for filter design. Secondly, in projector stabilizing applications, system errors can significantly affect the projected video experience since a small error in angle will result in a larger offset in projected pixel positions. Thus, accuracy in the short term is our second most important specification. Finally, drift over long term has been evaluated on our SENSE block by monitoring the outputs for a sufficient amount of time for a short presentation.

In this fusion block design, the 3-axis gravity measurements from the accelerometer are fused with the compensated magnetic field data in the first stage then fused with the gyroscope data in the second stage. The ellipsoid compensation block is used to map and normalize the raw magnetic field data from Earth's ellipsoid magnetic shape to a spherical unity scale. The compensated magnetic field data is then fused with the accelerometer data.

For this first stage, there are many well developed algorithms that can be used to fuse these two parameters. These algorithms can be divided into two categories -- non-optimal and optimal algorithms. A popular method is the optimal algorithm category with various recursive estimation methods, Ref. [55]. The QUEST algorithm in Ref. [53] (and recently an improved version in Ref. [54]) has been used popularly in many navigation applications. While this algorithm performs better than the non-optimal one in converging the estimation errors, its speed is not fast enough for the response of our compensating requirement, Ref. [55]. Thus, instead of using the QUEST algorithm, we used the TRIAD algorithm, Ref. [57], for more robustness and smaller latency. The outputs of this block then are fused with the angular rate measurements from the gyroscope in the next Kalman filter block. The quaternion rotation vectors are used instead of Euler angles to prevent the singularity problem, Ref. [38][39]. Details about quaternion and its rotation representation can also be found in Ref. [41]. Note that all sensors that have different units have to be normalized before being fused by the Kalman filter block in the next step. For this estimator, as mentioned in the related work, *section 5.2*, there are two approaches to the filter designs for this block -- the complimentary filter approach and the Kalman filter approach. The former is a simpler, faster, and less computational resource required type. However, it does not perform well in the long term in noise environment due to its inability to eliminate the system errors and sensor noises. The Kalman filter performs better in this aspect but requires an expensive matrix inversion operation at every estimating point. The filter type is dependent more on the application requirement. For our filter design, a simplified version of the well-known complimentary Kalman filter has been implemented in which the inverse matrix is simplified and the quaternion representation has been used for rotation operation. We report that the compensation factor is optimized to 0.65-0.75 to get the lowest latency. For the scalar projection, we filtered out the bias by the reference block to map the Earth's coordinate system to a 2-D projection screen. Note that the data from low-cost MEMS accelerometers cannot be double-integrated for an extended period of time to determine position, due to a quadratic growth of errors, Ref. [34]. We verified this quadratic growth of errors with double-integration in our experimental system. Thus, this is a limitation of this IAPS approach at this time so we report that the IAPS does not have the same level of accuracy pixel compensation as with the CAPS approach during long term operation. Following are sections of implementation and result reports for IAPS on the projected video with comparison of two cases *with* and *without* compensations.

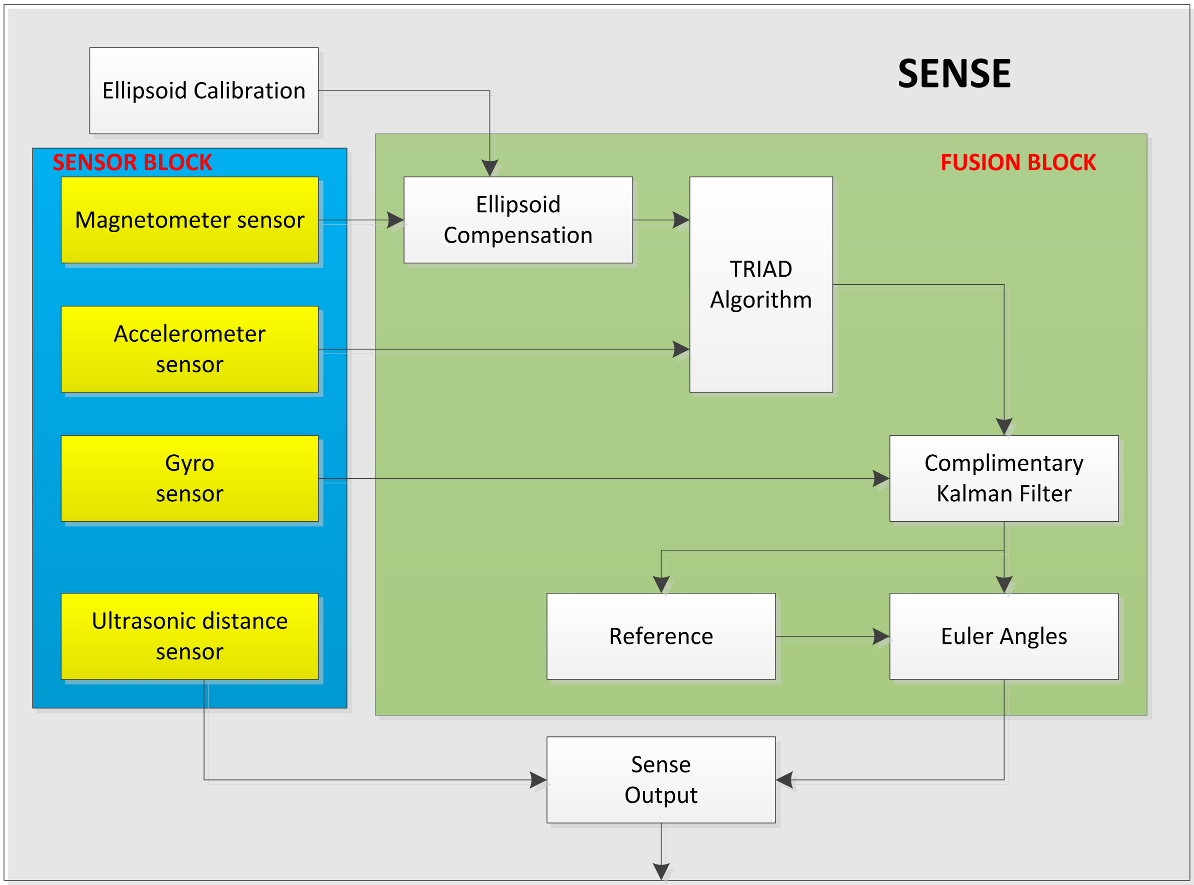


Figure 5.5 IAPS SENSE block diagram

5.7 Rotation Compensation in a Mobile Device

5.7.1 Stabilizing Projected Image with Roll Rotation in a Mobile Device

Figure 5.6 shows the fused angle output θ_z of a projector in Z axis with a slow roll movement. Note that in Figures 5.6 and 5.7, the red line represents the fused angle output; and the blue line represents the gyroscope raw data of the same orientation. Also, the horizontal grid scale is 100ms/unit and the vertical grid scale is 1 rad/unit. The slope changes of the projector's angle output line up against the gyroscope data as well as with the movement's start and stop showing that the response of the IMU system is accurately synchronous with the rotation movement of the projector. This is a major requirement for fast movement to prevent projected image shaking. Figure 5.7 shows the fused angle output of the projector in Z axis with fast changing speeds in roll movement. The angle changes according to the fast movement show that the response of the IMU system is accurately following the device movement simultaneously. Based on the fused angle output θ_z of the projector in Z axis, the compensation matrix of projected image can be derived as:

$$C_r = \begin{pmatrix} \cos\theta_z & \sin\theta_z & 0 \\ -\sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.14}$$

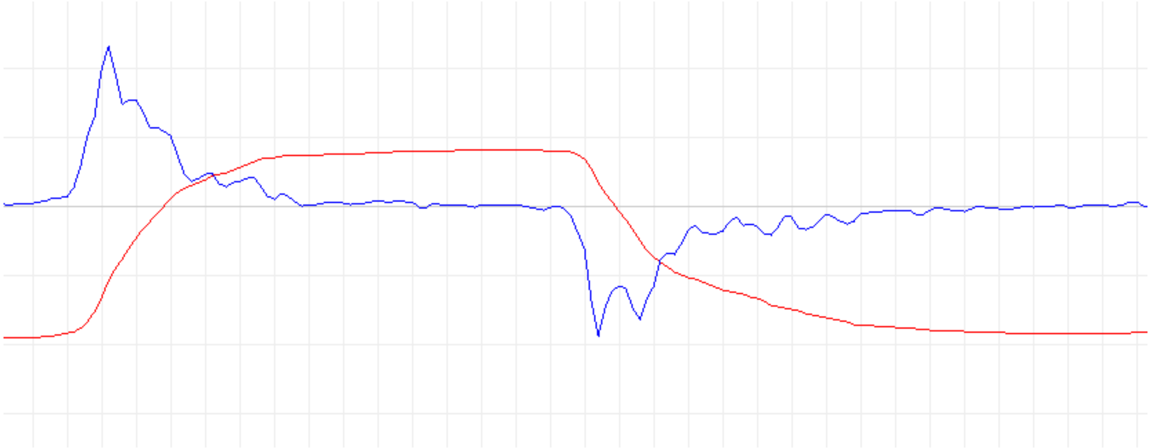


Figure 5.6   Roll Output with Projector Slow Movement

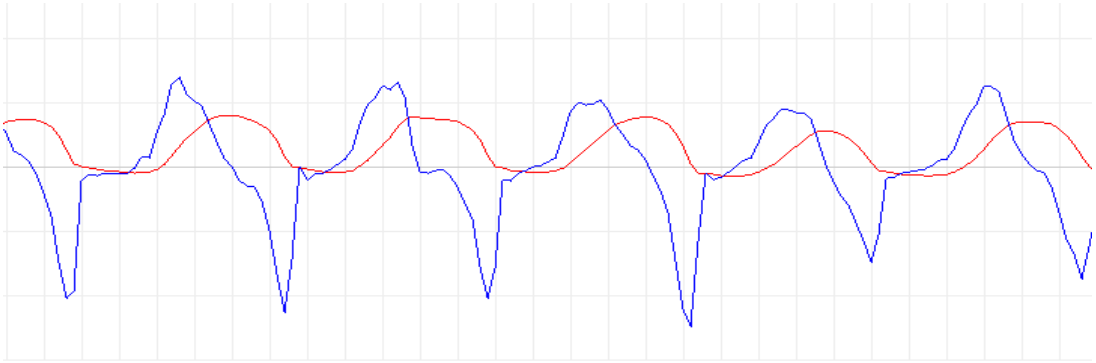


Figure 5.7 Roll Output with Projector Fast Movement

Figure 5.8 is the snapshot of rolling movement with (right) and without (left) stabilizing compensation. Note that on the left side of the figure, without stabilization, the text on rolling projector was distorted and hard to read. On the right side of the figure, with stabilization, the text was compensated and kept the same straight position as with the original text. The recorded video link of this snapshot is listed in Ref. [43].

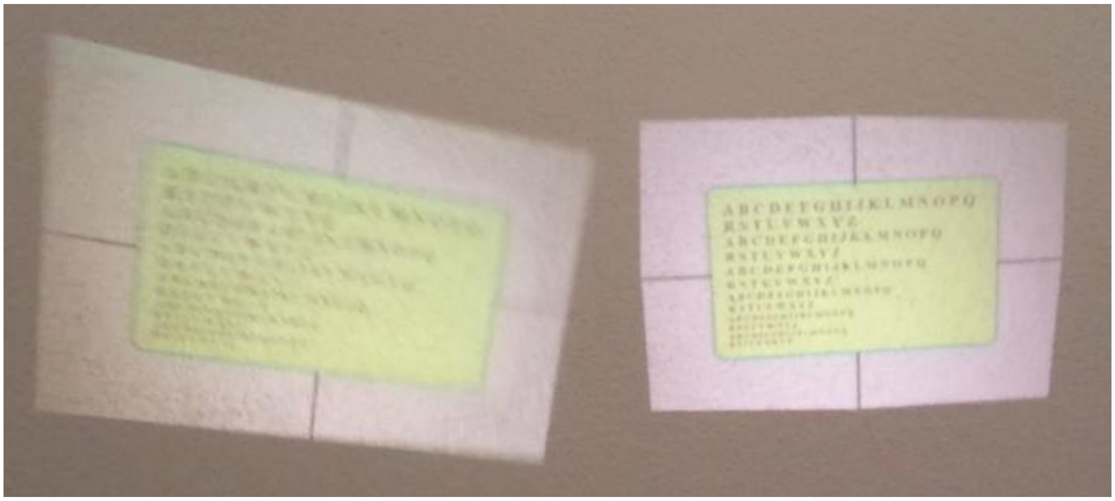


Figure 5.8 Projected Video Snapshot of Roll Rotation

5.7.2 Stabilizing Projected Image with Yaw Rotation in a Mobile Device

Similarly, Figure 5.9 shows the fused angle output θ_x of the projector in X axis with slow yaw movement. The slope changes also line up against the gyroscope data along the movement's start and stop, again showing that the response of the IMU system is accurately synchronous with the projector's yaw movement. This is an even more stringent requirement for the projector stabilization because any small angle movement results in a large offset of the horizontal displacement. Figure 5.10 shows the fused angle output of the projector in X axis with fast changing speeds in yaw movement. The angle changes show that the response of the IMU system is accurately following the device movement in a horizontal direction at the same time as well.

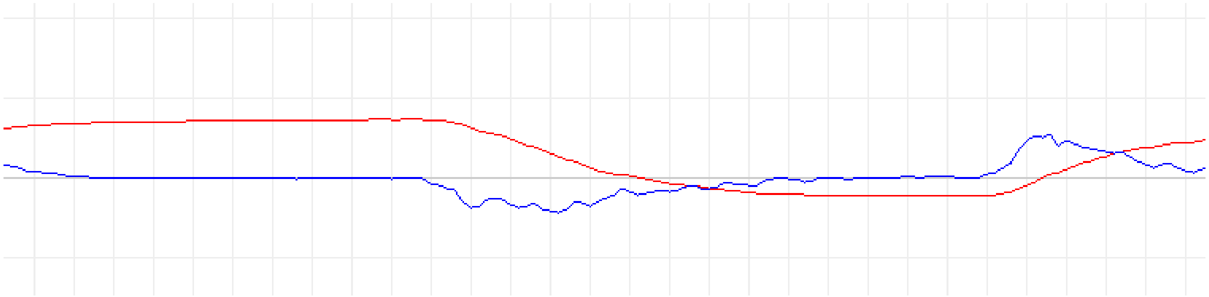


Figure 5.9 Yaw Output with Projector Slow Movement

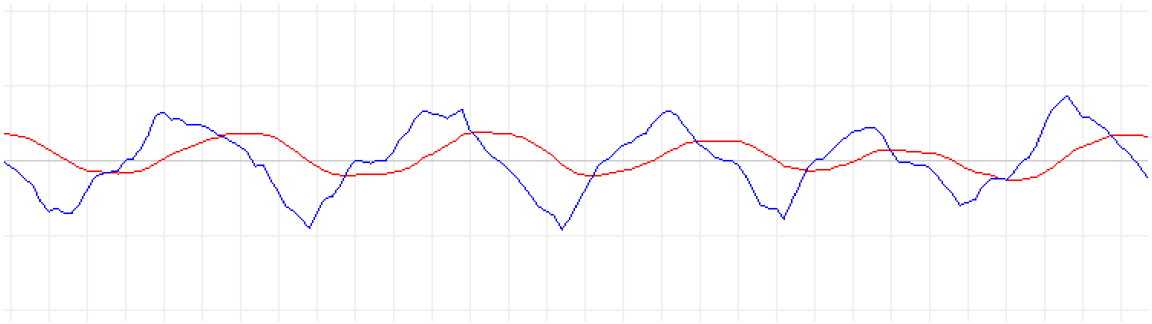


Figure 5.10 Yaw Output with Projector Fast Movement

Based on the fused angle output θ_x of the projector movement in X axis, the compensation matrix of the projected image can be derived as:

$$C_x = (I|A_x) \tag{5.15}$$

With I as identity matrix 3x3, and A_x derived from:

$$A_x = [p_x * r_x * \tan(\theta_x) \;\; 0 \; 0 \; 1]^T \tag{5.16}$$

And with p_x as the projector parameter that we get from each projector's calibration result at once, r_x is the screen ratio factor constant. Figure 5.11 is the snapshot of yaw movement with (right) and without (left) stabilizing compensation. Without stabilization on the left side of the figure, the text on the moved projector was distorted and blurred. The recorded video link of this snapshot is posted in Ref. [44]. Measuring the gravity vector in the sensor coordinate frame using accelerometers allows estimation of the orientation relative to the horizontal plane. However, when the sensor module is rotated about the vertical axis in the yaw movement, the gravity vector on each of the principal axes of the accelerometer will not change due to the fact that the accelerometer cannot sense the rotation about the vertical axis. The magnetometer data is then used solely in this case to measure the local magnetic field vector in sensor coordinates and allows the determination of orientation relative to the vertical.

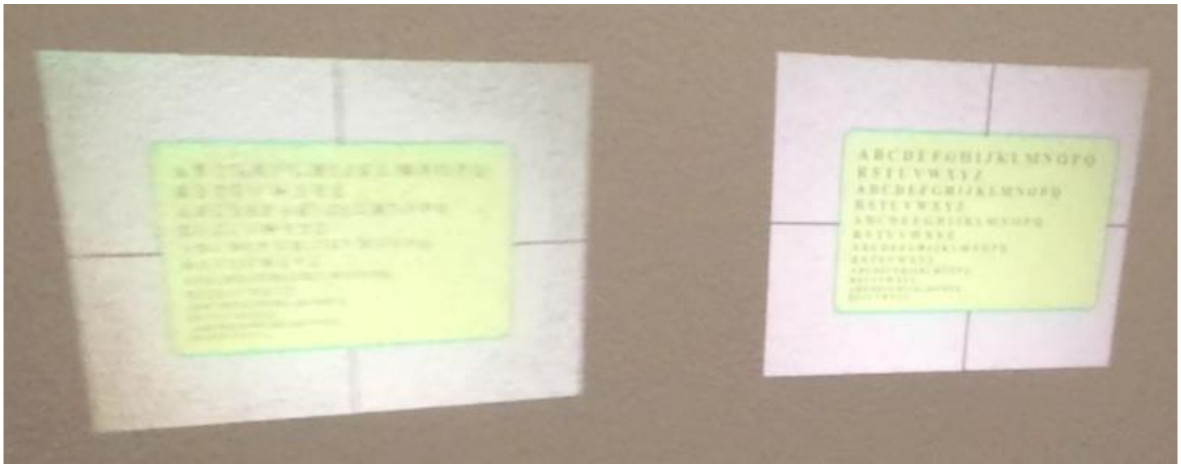


Figure 5.11 Projected Video Snapshot of Yaw Movement

5.7.3 Stabilizing Projected Image with Pitch Rotation in a Mobile Device

In the same analysis as with other movements, Figure 5.12 shows the fused angle output θ_y of the projector in Y axis with slow pitch movement. Again, the angle slope changes of the projector also line up against the gyroscope data as well as the movement's start and stop, showing that the response of the IMU system is accurately synchronous with the pitch movement of the projector. This is also a stringent requirement for the projector stabilization because any small angle movement will result in a large offset of the vertical displacement. Figure 5.13 shows the fused angle output with fast changing speeds in pitch movement. These fast changes show that the response of the IMU system is accurately following device movement in vertical axis as well. Although the pitch rotation angle is translated to the displacement in vertical orientation in the same manner as with the yaw rotation, the displacement in vertical ratio is different from the horizontal accordingly in relation to their screen aspect. This ratio has to be taken into the compensation matrix with the screen ratio factor constant r_y as follows:

$$C_y = \left(I|A_y\right) \tag{5.17}$$

$$A_y = [p_y * r_y * \tan\left(\theta_y\right)\ 0\ 0\ 1]^T \tag{5.18}$$

With p_y as the projector parameter that we get from each projector calibration result, r_y is the screen ratio factor constant. Figure 5.14 is the snapshot of pitch movement with (right) and without (left) stabilizing compensation. On the left side, without stabilization, the text on the moved projector was distorted and blurred. The recorded video link of this snapshot is posted in Ref. [45].

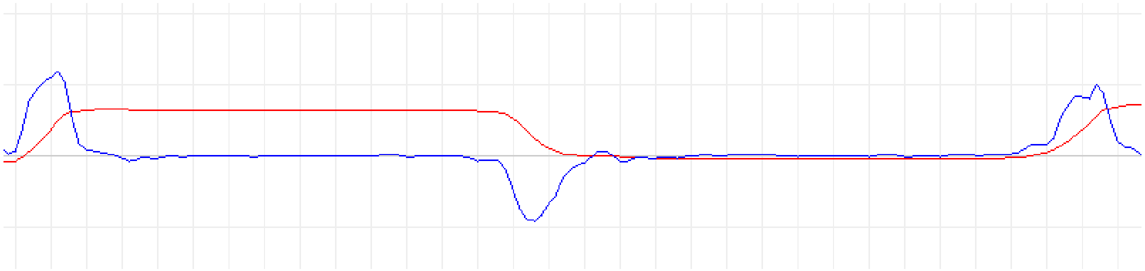


Figure 5.12 Pitch Output with Projector Slow Movement

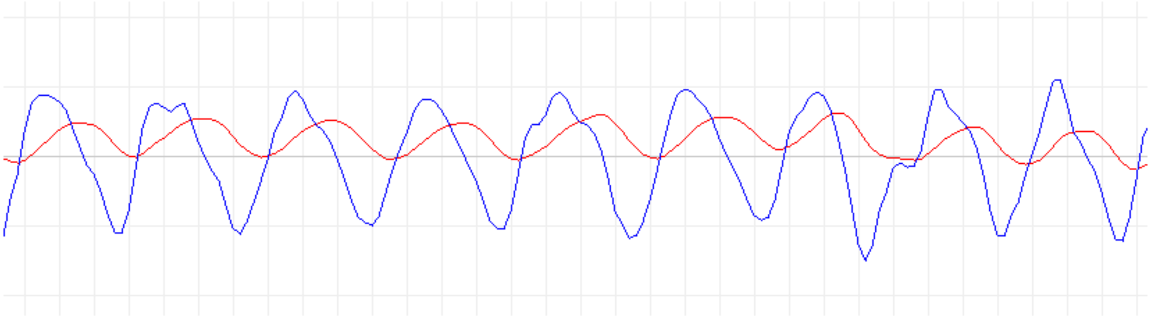


Figure 5.13 Pitch Output with Projector Fast Movement

5.7.4 Stabilizing Projected Image with Combined Rotation in a Mobile Device

The combined roll, yaw, and pitch movement of the projector are translated and compensated in the composition of the compensation matrices:

$$C_c = C_r * C_y * C_x \tag{5.19}$$

with C_c is the combined compensation matrix for these operations, C_r, C_y, C_x are compensation matrices for roll, pitch, and yaw rotations respectively.

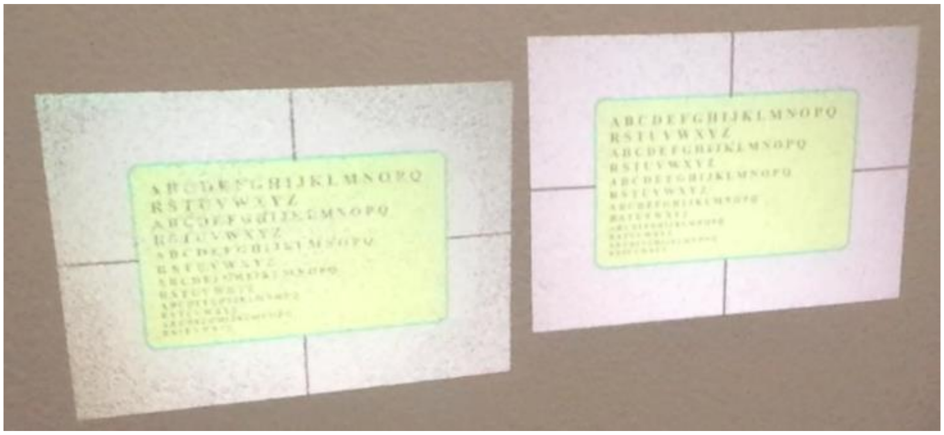


Figure 5.14 Projected Video Snapshot of Pitch Movement

Figure 5.15 is the snapshot of combined movement with (right) and without (left) stabilizing compensation. On the left, without stabilization, the text was distorted and hard to read. With stabilization, the text on the right side of the figure was compensated to keep the same straight position as in the original text. The recorded video link of this snapshot is posted in Ref. [46].

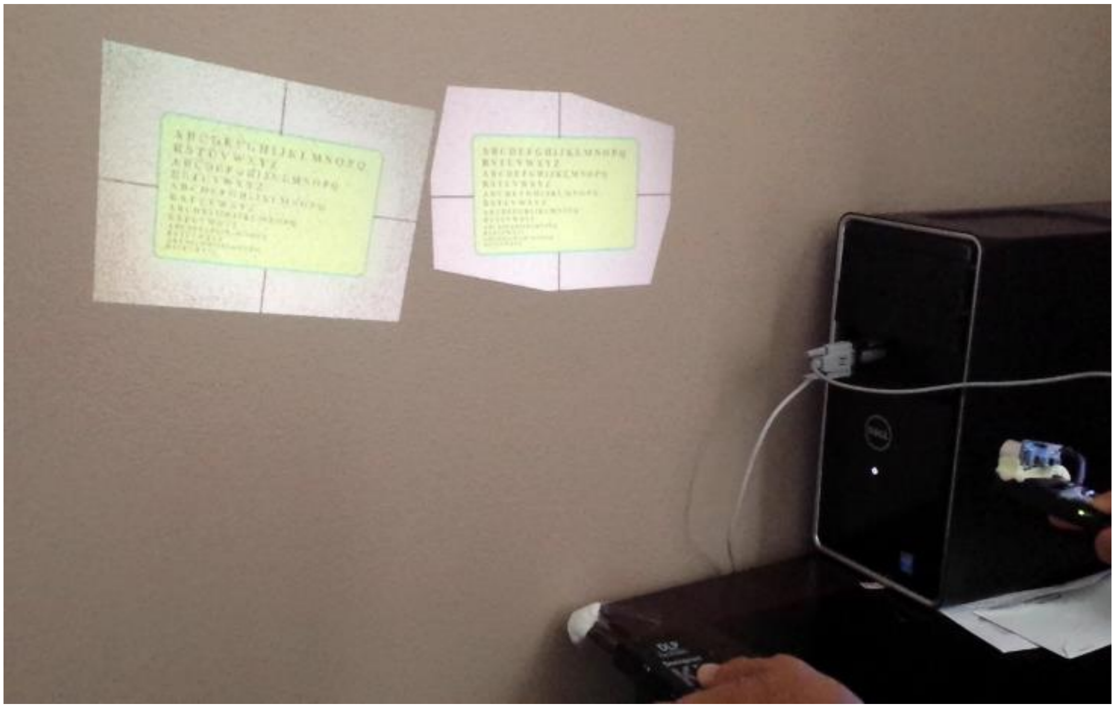


Figure 5.15 Projected Video Snapshot of Combined Rotation

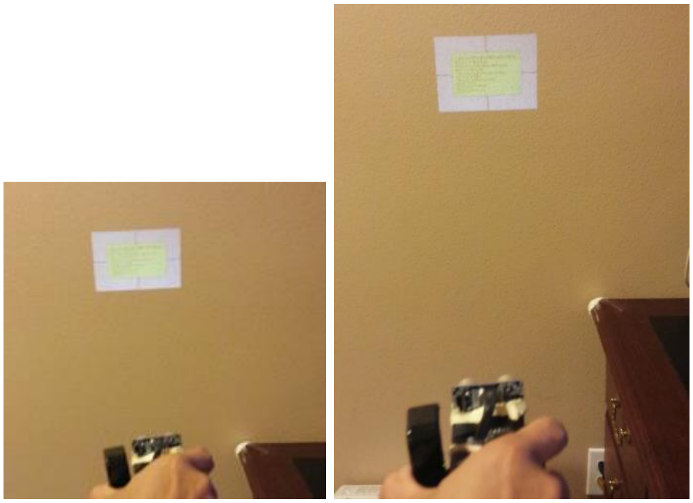
5.7.5 Distance Compensation in a Mobile Device

When zooming in or out, or simply moving the projector forward or backward, the projected screen image size changes with the distance between the projector and the screen. In order to keep the screen image size the same as the original projected image, we scaled the image with the current measured distance and the reference measured distance.

We used a generic common distance sensor, HC-SR04, Ref. [56], for our prototype to provide a distance measurement from 2cm to 400cm with an accuracy of 3mm which is a good range, enough for a typical pico projector distance range. In a practical usage scenario, the user presses the control key to measure reference distance d_r. After that, the distance variation x is monitored continuously during the presentation. The pre-defined zoom fraction r is an optional choice for selecting a user's desired zoom factor. Finally the scaling factor $s(x)$ is used to keep the projected video size constant:

$$s(x) = d_r/(x * r) \tag{5.20}$$

To filter out the outliers, we used a linear regression method to check each distance measurement before updating with the validated stored distance measurements. The scale factor is updated in real time at 100Hz rate which is greater than the typical refresh rate of the projected video (60Hz) to eliminate the artifact of the zoom effect due to changes of distance. The evaluation of the stabilized video with the distance variation has shown that the projected video size is stabilized with no noticeable jagging during our observation. Figure 5.16(a) and 5.16(b) are the snapshots of changing the distance to about 1 foot from a fixed projected video size. The recorded link of the snapshot is posted in Ref. [42].



(a) close position (b) far position

Figure 5.16 Projected Image Size – zoom (a) close (left) and (b) far (right) positions

5.7.6 Quantifying compensation results

To quantify the compensation effect, we projected a reference cross line that will not be compensated to compare with the compensated corresponding video image which is the same cross line. Then we setup a system as shown in Figure 5.17 to measure the displacements before and after compensation for the yaw, pitch, roll and zoom movements.

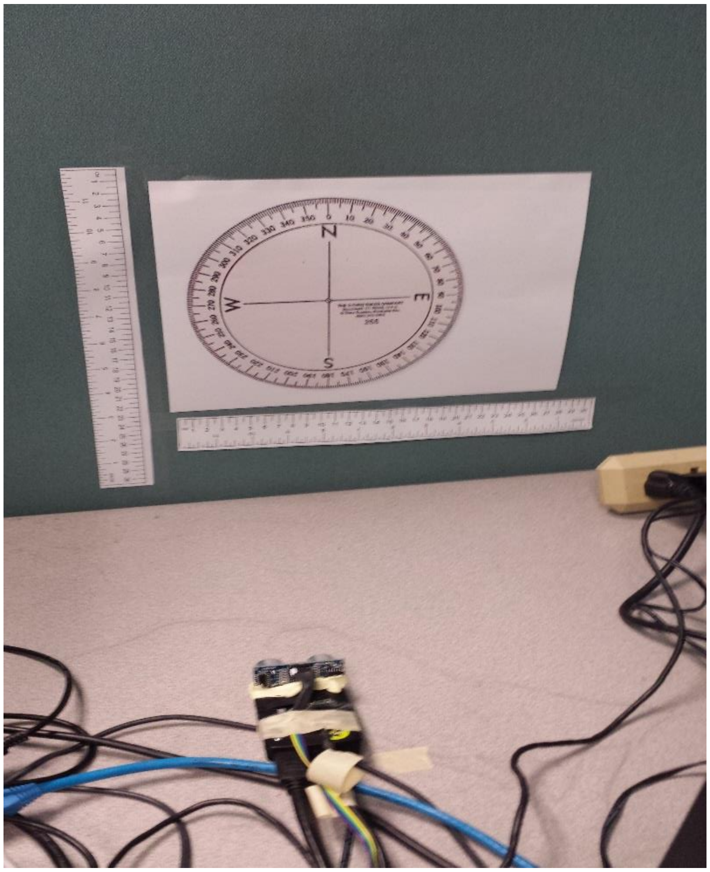


Figure 5.17 System setting for quantifying compensation results

For the rotation movement, we rotated the projector from 0 degree to 90 degree to measure the rotation errors after compensated. We noticed that the compensated cross line stays about 0 degree during the rotation which is perfectly compensated. Figure 5.18 shown the compensated cross line stays about 0 degree when the reference cross line rotated to 50 degree.

For the yaw and pitch movements; however, when apply the similar method to quantify the compensations of yaw and pitch movements, we can measure some compensated yaw and pitch displacement values. Table 5.1 lists the displacement values of non-compensated and compensated cases. Also Figure 5.19 represents the displacements on those yaw and pitch movements. The projector throw distance is 35 cm. The displacement measurements are performed until the video image being out of range to compensate by warping video image.

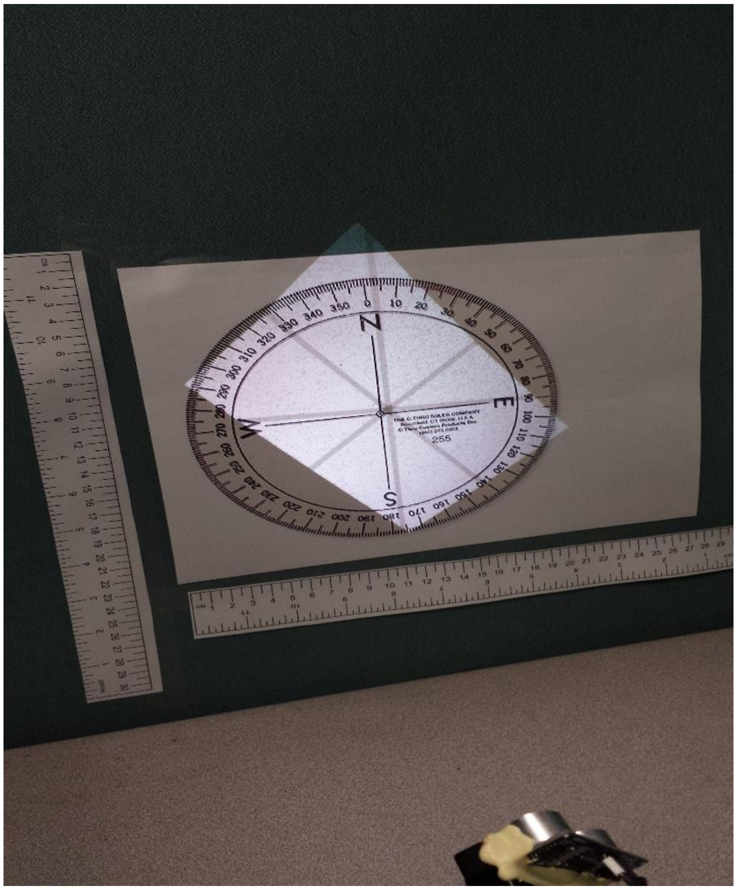


Figure 5.18 Projected image with and without roll compensation

Table 5.1 Displacement with and without yaw and pitch compensations

Non-Compensated Yaw/Pitch Displacement (cm)	Non-Compensated Yaw/Pitch Displacement Angle (degree)	Compensated Yaw Displacement (cm)	Compensated Yaw Displacement Angle (degree)	Compensated Pitch Displacement (cm)	Compensated Pitch Displacement Angle (degree)
0	0	0	0	0	0
2	3.3	0.1	0.2	0.1	0.2
4	6.5	0.2	0.3	0.3	0.5
6	9.7	0.4	0.7	0.6	1
8	12.9	0.6	1	0.8	1.3
10	15.9	0.8	1.3	out of screen	
12	18.9	1	1.6		
14	21.8	out of screen			

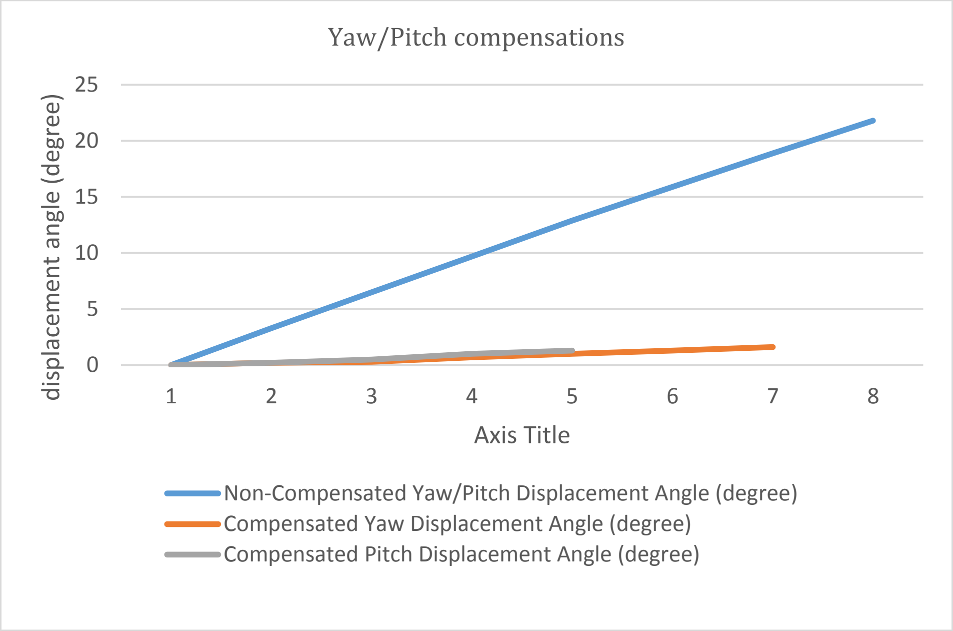


Figure 5.19 Yaw/Pitch compensations – non-compensated and compensated displacement values

Table 5.2 and 5.3 shows the projected video size before and after the zoom compensation. Also, Figure 5.20 shows the differences in term of projected video sizes to the throw distance with and without compensated accordingly.

Table 5.2 Projected Video Size without compensation

Distance (cm)	Distance Ratio	Compensation Factor	Projected Video Sized (cm)	Projected Video Sized (%)
15.3	0.44	1	1.6	35.56%
16.6	0.47	1	2	44.44%
18.61	0.53	1	2.3	51.11%
21.49	0.61	1	2.8	62.22%
23.91	0.68	1	3.2	71.11%
28.18	0.81	1	3.7	82.22%
32.09	0.92	1	4.2	93.33%
34.42	0.98	1	4.5	100.00%
37.13	1.06	1	4.8	106.67%
40.35	1.15	1	5.2	115.56%
43.13	1.23	1	5.5	122.22%
46.67	1.33	1	6.2	137.78%
50.34	1.44	1	6.7	148.89%
54.48	1.56	1	7.1	157.78%
58.53	1.67	1	7.5	166.67%
65.68	1.88	1	8.2	182.22%
67.04	1.92	1	8.5	188.89%

Table 5.3 Projected Video Size with zoom compensation

Distance (cm)	Distance Ratio	Compensation Factor	Projected Video Sized (cm)	Projected Video Sized (%)
15.64	0.45	2.24	4.45	98.89%
16.45	0.47	2.13	4.5	100.00%
22.02	0.63	1.59	4.5	100.00%
25.81	0.74	1.36	4.55	101.11%
30.78	0.88	1.14	4.55	101.11%
35.45	1.01	0.99	4.5	100.00%
39.79	1.14	0.88	4.53	100.67%
43.73	1.25	0.8	4.5	100.00%
48.88	1.4	0.72	4.55	101.11%
53.95	1.54	0.65	4.55	101.11%
60.83	1.74	0.58	4.5	100.00%
67.76	1.94	0.52	4.5	100.00%
68.29	1.95	0.51	4.5	100.00%

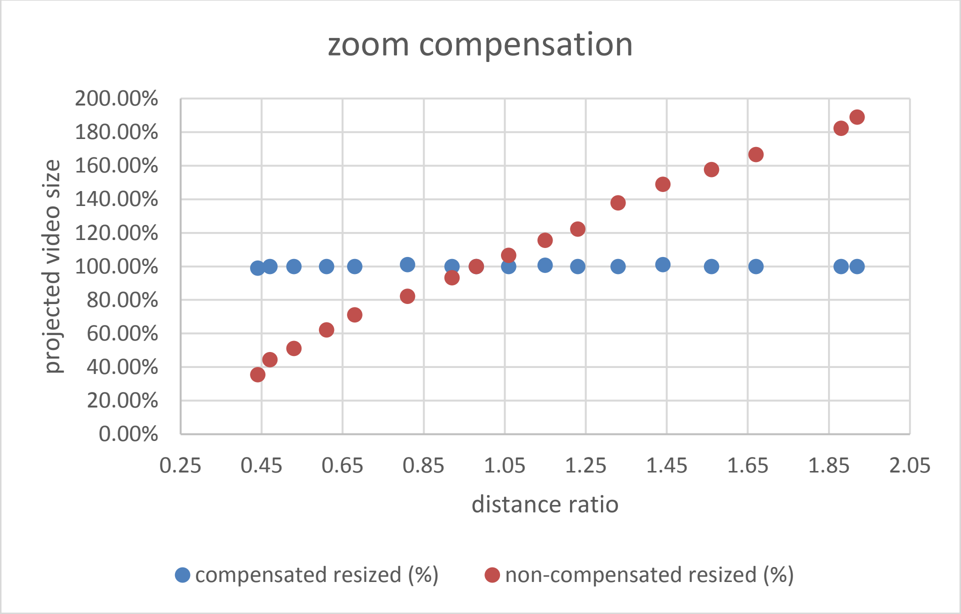


Figure 5.20 Comparison of Projected Video Size with and without zoom compensation

5.8 Conclusion

Two methods of stabilizing the portable projector enabled mobile device were presented in this chapter. The first method, named Camera-based Assisted Projector Stabilization method (CAPS), detects the different reference patterns on the captured images to derive the compensation matrix, then to use it to warp the video image to render the same projected image on the screen. This is the first attempt to do these tasks in real time and proved to compensate for slower motions. In the non-embedded projector mobile device setting, while a mobile device is connected to a stand-alone pico projector, this method is simple for typical smart phone on the market. In the embedded projector mobile device, however, the requirement to have fixed passive markers on the screen can cause inconvenience to users. To eliminate that inconvenience, the second method, IMU Assisted Projector Stabilization method (IAPS), seems to be more preferable. Also this IAPS method responded better to faster motion and proved to be able to compensate in all rotation movements. Projector stabilization applications can be useful for reading projected text, for social events, for quick presentations, and for hand-held gaming environments. This can open doors to more research in projector stabilization to have better video (i.e. working on faster hardware platforms and more accurate and less noise sensors). Finally, it is worthwhile to pursue motion tracking with linear displacement by adding more sensors to track motion and/or develop better estimator algorithms and faster warping processing algorithms.

Bibliography

[1] Pablo Roman, Maxim Lazarov, and Aditi Majumder, "A Scalable Distributed Paradigm for Multi-User Interaction with Tiled Rear Projection Display Walls," *IEEE Transactions on Visualization and Computer Graphics,* 2010.

[2] K. Amiri, Shih Hsien Yang, C. Larsen, F. Kurdahi, M. El Zarki, A. Majumder, "Camera-based video synchronization for a federation of mobile projectors," *Computer Vision and Pattern Recognition Workshops (CVPR)*, 2011 IEEE Computer Society Conference.

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

[4] 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.

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

[6] A. Raij, G. Gill, A. Majumder, H. Towles, and H. Fuchs, "PixelFlex2: A Comprehensive, Automatic, Casually-Aligned Multi-Projector Display," *IEEE PROCAMS*, 2003.

[7] 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.

[8] 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.

[9] T. OKATANI, K. DEGUCHI, "Autocalibration of a Projector-Screen-Camera System: Theory and Algorithm for Screen-to-Camera Homography Estimation," *IEEE ICCV*, 2003.

[10] B. SAJADI, A. MAJUMDER, "Markerless View Independent Geometric Registration of Multiple Distorted Projectors on Vertically Extruded Surfaces Using a Single Uncalibrated Camera," *IEEE Transactions on Visualization and Computer Graphics*, 2009.

[11] A. MAJUMDER, B. SAJADI, "Advances in Large Area Displays: The Changing Face of Visualization," *IEEE Computer*, 2013.

[12] DLP Pico Projector Kit

http://www.ti.com/lsds/ti/analog/dlp/tools-software.page#evm

[13] Prototype microprocessor board

https://www.raspberrypi.org

[14] openMAX specification link

https://www.khronos.org/openmax

[15] openGL ES specification link

https://www.khronos.org/opengles

[16] DUY QOC-LAI, BEHZAD SAJADI, SHAN JIANG, ADITI MAJUMDER, M. GOPI, "A Distributed Memory Hierarchy and Data Management for Interactive Scene Navigation and Modification on Tiled Display Walls," *IEEE TVCG*, 2014.

[17] RICHARD HARTLEY AND ANDREW ZISSERMAN, "Multiple View Geometry in Computer Vision," *Cambridge University Press*, 2004.

[18] https://en.wikipedia.org/wiki/Eye#Visual_acuity

[19] B. JOHANSSON, "View synthesis and 3D reconstruction of piecewise planar scenes using intersection lines between the planes", *Proc. 7th Int. Conference on Computer Vision,* pp.54-59, 1999

[20] O. FAUGERAS, "Three-Dimensional Computer Vision: A Geometric Viewpoint" *MIT Press,* Cambridge, Massachusetts, 1993

[21] R. HARTLEY, "In defense of the eight-point algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol.19, No.6, 1997

[22] R. SUKTHANKAR, R. STOCKTON, AND M. MULLIN, "Smarter presentations: Exploiting homography in camera-projector systems", *Proc. ICCV,* 2001

[23] R. RASKAR, "Immersive planar display using roughly aligned projector", *IEEE Virtual Reality,* 2000

[24] R.RASKAR, J. CHAI, AND J. VANBAAR, "Low cost projector mosaics", *Proc. of Fifth Asian Conference on Computer Vision,* 2002

[25] RICHARD HARTLEY AND ANDREW ZISSERMAN, "Multiple View Geometry in Computer Vision", *CAMBRIDGE University Press,* 2004

[26] RAMESH RASKAR AND PAUL BEARDSLEY, "A Self-Correcting Projector", *IEEE Computer Vision and Pattern Recognition (CVPR),* 2001

[27] LIANG YAN-DE, CHENG MIN, HE FU-BEN, AND LI HANG, "Attitude estimation of a quadrotor aircraft based on complementary filter", *Transducer and Microsystem Technologies,* 2011

[28] JAMES CALUSDIAN, XIAOPING YUN, AND ERIC BACHMANN, "Adaptive-gain complementary filter of inertial and magnetic data for orientation estimation", *IEEE, editor. IEEE ICRA,* 2011

[29] SEBASTIAN O. H. MADGWICK, ANDREW J. L. HARRISON, AND RAVI VAIDYANATHAN, "Estimation of IMU and MARG orientation using a gradient descent algorithm", *IEEE, editor. IEEE ICORR,* 2011

[30] XIAOPING YUN, ERIC R. BACHMANN, AND ROBERT B. MCGHEE, "A Simplified Quaternion-Based Algorithm for Orientation Estimation From Earth Gravity and Magnetic Field Measurements", *IEEE Transactions on Instrumentation and Measurement,* Vol. 57, No. 3, 2008

[31] J. L. MARINS, X. YUN, E. R. BACHMANN, R. B. MCGHEE, AND M. J. ZYDA, "An extended Kalman filter for quaternion-based orientation estimation using MARG sensors," *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.,* 2001, pp.2003–2011

[32] LI WANG, ZHENG ZHANG AND PING SUN, "Quaternion-based Kalman Filter for AHRS Using an Adaptive-step Gradient Descent Algorithm", *International Journal of ARS,* 2015

[33] E. R. BACHMANN, I. DUMAN, U. Y. USTA, R. B. MCGHEE, X. P. YUN, AND M. J. ZYDA, "Orientation Tracking for Humans and Robots Using Inertial Sensors", *Proc. of 1999 Symposium on CIRA,* 1999

[34] XIAOPING YUN AND ERIC R. BACHMANN, “Design, Implementation, and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking”, *IEEE Transaction on Robotics,* Vol. 22, No. 6, 2006

[35] E. FOXLIN, “Inertial head-tracker fusion by a complementary separate bias Kalman filter,” *In Proc. Virtual Reality Annu. Int. Symp.,* 1996, pp.185–194

[36] A.M SABATINI, **“**Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing”, *IEEE Transactions on Biomedical Engineering,* 2006

[37] A.M. SABATINI, **“**Kalman-Filter-Based Orientation Determination Using Inertial/Magnetic Sensors: Observability Analysis and Performance Evaluation”, *Sensors,* 11(10): 9182-206, 2011

[38] J.M. COOKE, M.J. ZYDA, D.R. PRATT, AND R.B. MCGHEE, “NPSNET: Flight Simulation Modeling Using Quaternions”, *Presence,* Vol.1, No.4, 1992

[39] A. ALAIMO, V. ARTALE, C. MILAZZO AND A. RICCIARDELLO, “Comparison between Euler and Quaternion Parametrization in UAV Dynamics”, *International Conference of Numerical Analysis and Applied Mathematics (ICNAAM),* 2013

[40] R. P. G. COLLINSON, **“**Introduction to Avionics Systems”, *3rd ed. Springer,* 2011. 530 p.

[41] JACK B. KUIPERS, **“**Quaternions and Rotation Sequences**”,** *Princeton University Press*, 1999

[42] RECORDED: changing distance from the screen https://www.youtube.com/watch?v=FRphf0eZwF8

[43] RECORDED: changing the roll orientation https://www.youtube.com/watch?v=PDeR4e8zYd4

[44] RECORDED: changing the yaw orientation

https://www.youtube.com/watch?v=K_kar31l5iU

[45] RECORDED: changing the pitch orientation

https://www.youtube.com/watch?v=NxikinaoSKw

[46] RECORDED: roll, yaw and pitch orientations

https://www.youtube.com/watch?v=d4B7v_xYi5c

[47] R. K. Lenz and R. Y. Tsai, "Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3-D Machine Vision Metrology", *IEEE TPAMI,* Vol. 10, No. 5, 1988, pp.713-720

[48] Ferdinand P. Beer and E. Russell Johnston, Jr, "Vector Mechanics for Engineers: Statics and Dynamics", *5th ed. McGraw-Hill,* 1988

[49] Donghui Liu, Ling Pe, Jiuchao Qian, Lin Wang, Chengxuan Liu, Peilin Liu, Wenxian Yu, "Simplified Ellipsoid Fitting-Based Magnetometer Calibration for Pedestrian Dead Reckoning", *CSNC2016 Proceedings,* Vol. II

[50] G. Grenon, P. E. An, S. M. Smith, and A. J. Healey, "Enhancement of the inertial navigation system for the morpheus autonomous underwater vehicles," *IEEE J. Ocean. Eng.,* Vol. 26, No. 4, pp.548-560, 2001

[51] E. R. Bachmann, R. B. McGhee, X. Yun, and M. J. Zyda, "Inertial and magnetic posture tracking for inserting humans into networked virtual environments," *Proc. ACM Symp. VRST,* 2001, pp.9-16

[52] A. Gallagher, Y. Matsuoka, and W.-T. Ang, “An efficient real-time human posture tracking algorithm using low-cost inertial and magnetic sensors,” *Proc. IEEE Int. Conf. Robot. Autom.,* 2004, pp.2967-2972

[53] M. Shuster and S. Oh, “Three-axis attitude determination from vector observations,” *Journal Guidance Control, and.Dynamics*, Vol. 4, No. 1, pp.70–77, 1981

[54] Yang Cheng and Malcolm D. Shuster, “An Improvement to the Implementation of the QUEST Algorithm”, *Journal of Guidance, Control, and Dynamics*, 2014

[55] F. Landis Markley and Daniele Mortari, “How to Estimate Attitude from Vector Observations”, *JAAS/AIAA Astrodynamics Specialist-Conference,* 1999

[56] Distance Sensor part specification link

www.micropik.com/PDF/HCSR04.pdf

[57] F. L. Markley, “Attitude Determination using Two Vector Measurements,” *Proceedings, Flight Mechanics Symposium*, 1999, NASA Conference Publication NASA/CP-19989-209235, pp. 39–52

[58] Gyroscope specification link

http://www.st.com/content/st_com/en/products/mems-and-sensors/gyroscopes/l3gd20h.html

[59] Compass specification link

http://www.st.com/content/st_com/en/products/mems-and-sensors/e-compasses/lsm303d.html

[60] HUNG NGUYEN, FADI KURDAHI, ADITI MAJUMDER, “Resource Aggregation for Collaborative Video from Multiple Projector enabled Mobile Devices”, *Proc. of the 14th ACM/IEEE Symposium on Embedded Systems for Real-Time Multimedia,* 2016

[61] Smart Mobile Device Users Estimation link

https://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694

[62] Mobile Video Traffic Estimation link

http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[63] RECORDED: Two unit Superimposed Setting Demo Link

https://youtu.be/ckvQuFO1wK8

[64] RECORDED: Four unit Tiled Setting Demo Link

https://youtu.be/2lJE5g10xfU

[65] RECORDED: Two unit Tiled Setting Demo Link

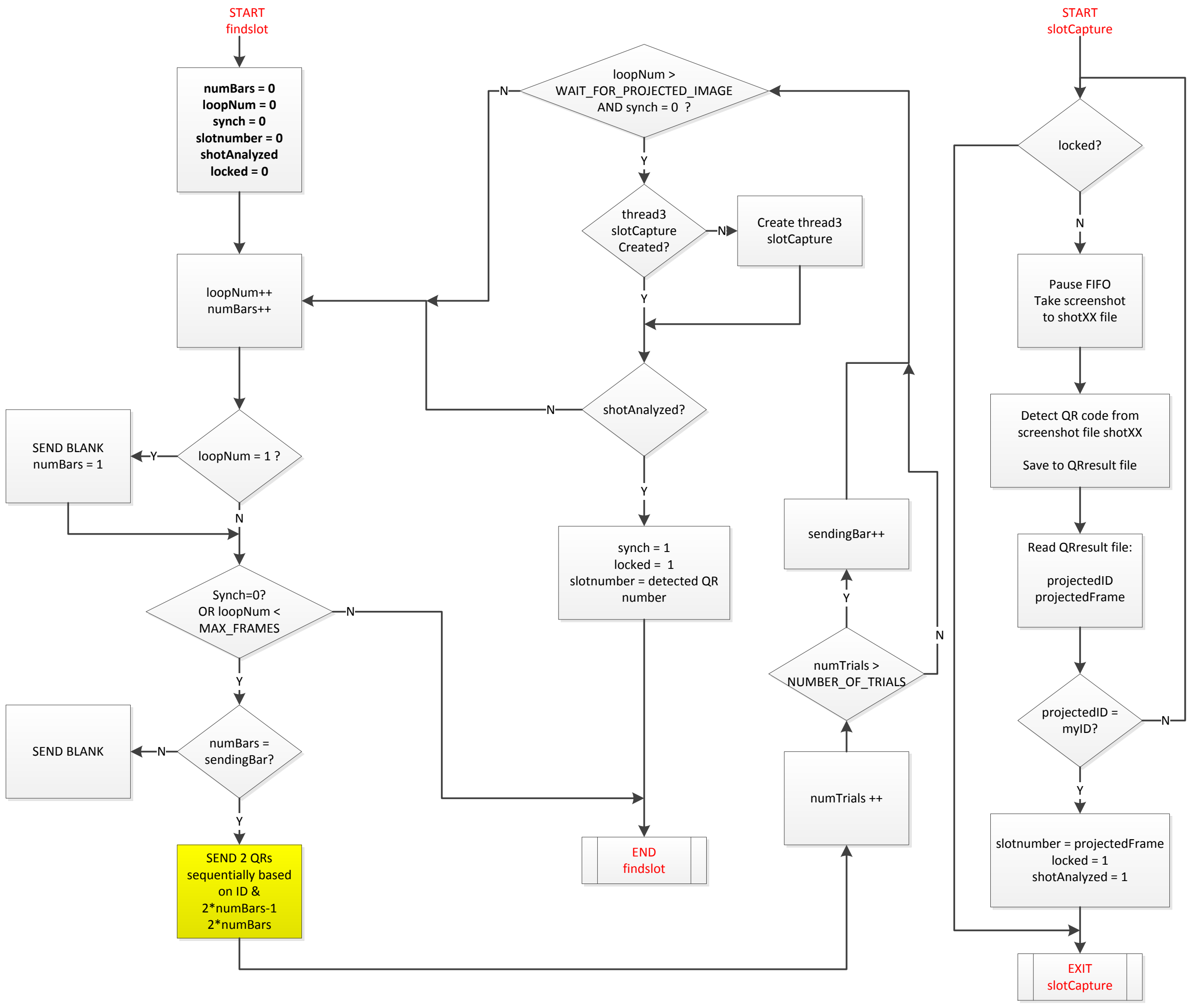
https://youtu.be/tsb1xs4XSA0

[66] RECORDED: Four unit Tiled Setting Re-Calibration Demo Link

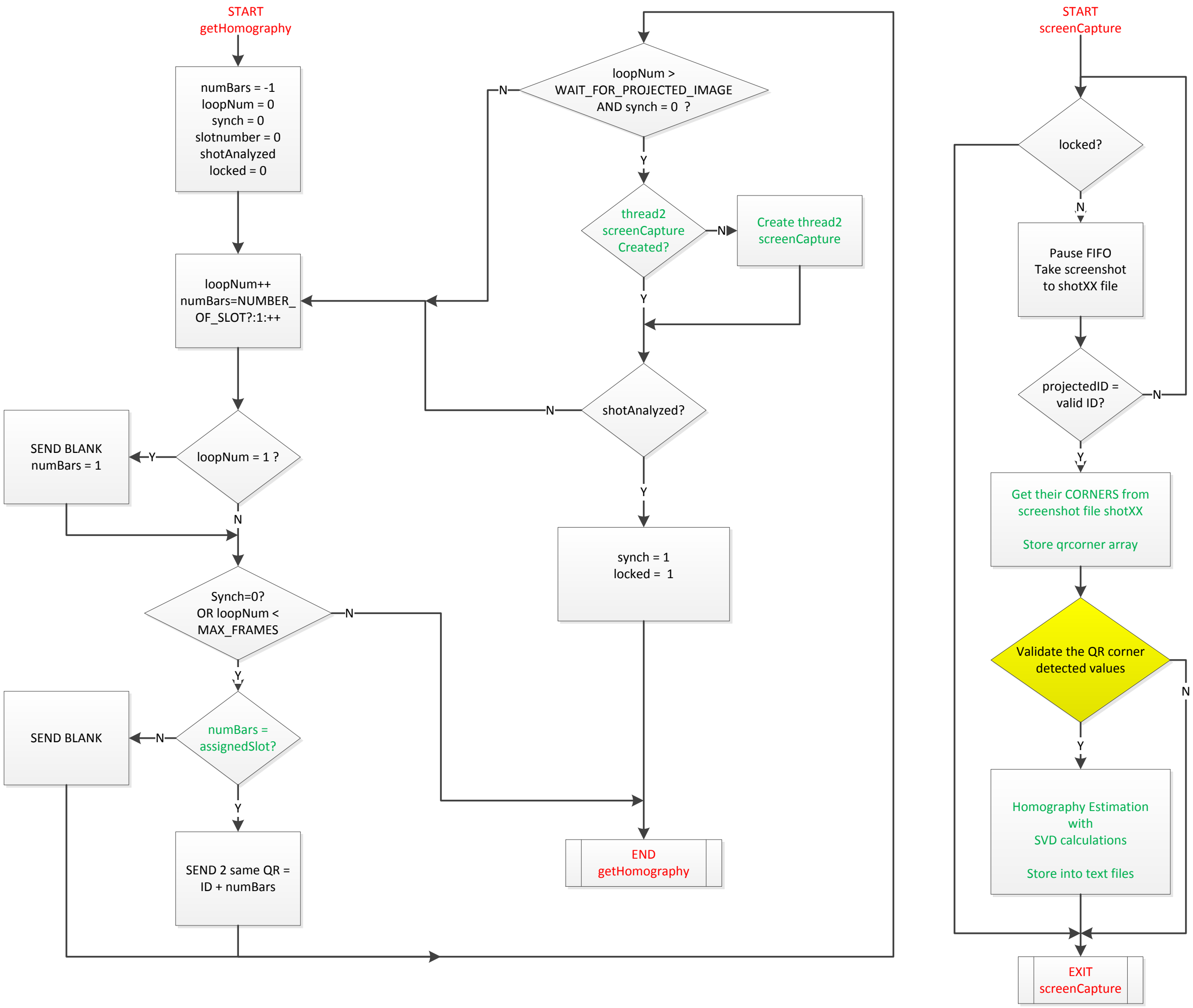
https://youtu.be/GyBMsEgklF8

Appendix

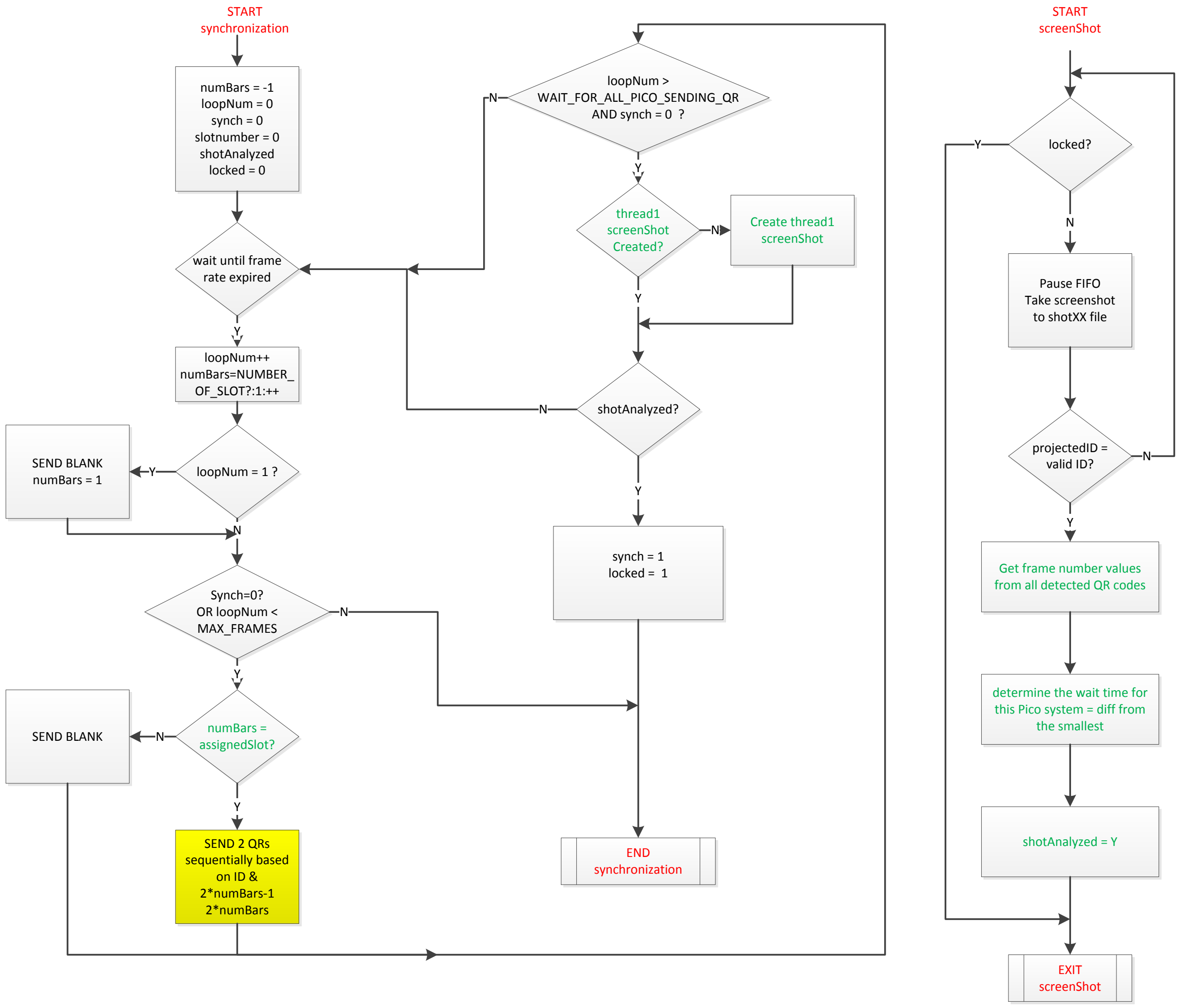
A Find Slot - Software Flow Chart



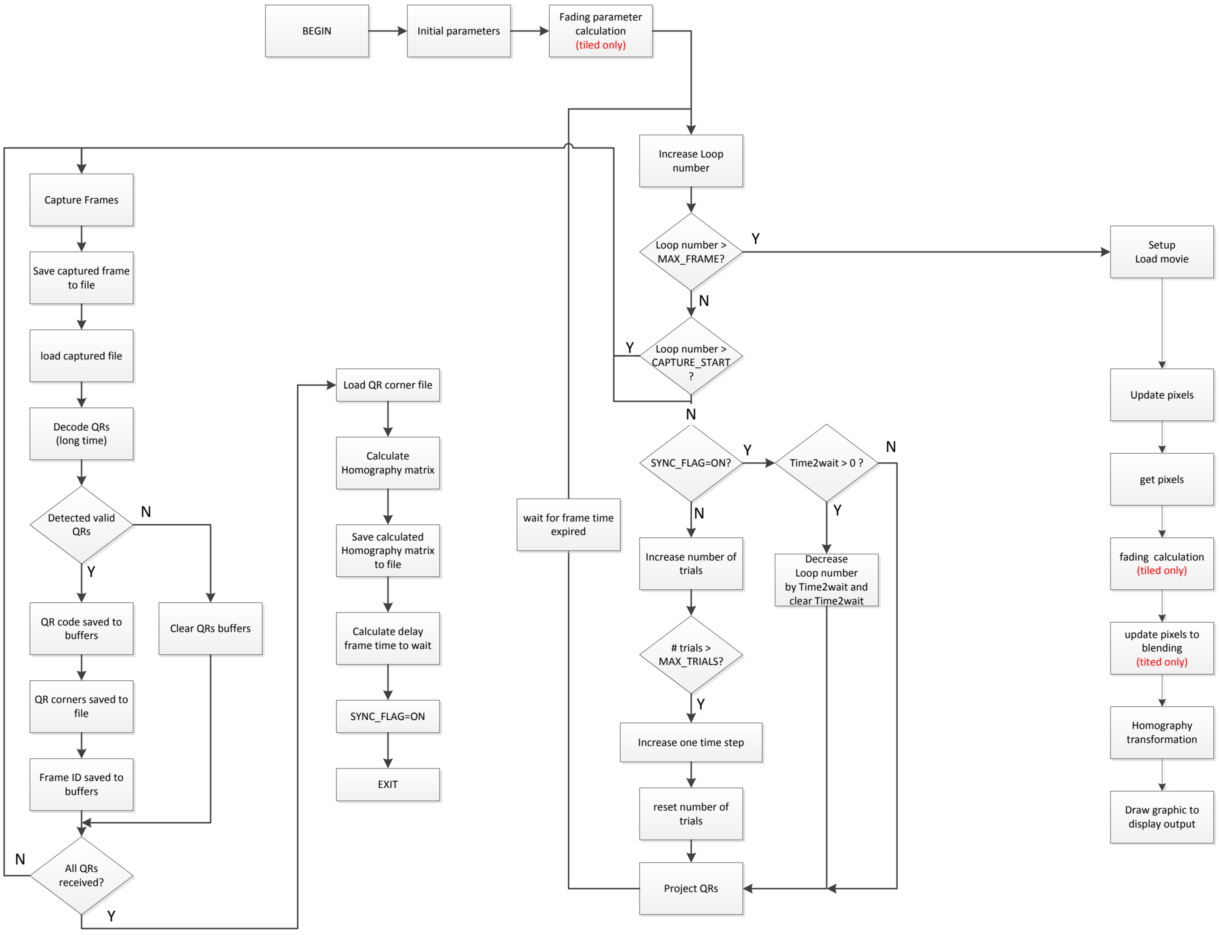
B Homography Detection Stage – Software Flow Chart



C Synchronization Stage – Software Flow Chart


START synchronization
numBars = -1
loopNum = 0
synch = 0
slotnumber = 0
shotAnalyzed
locked = 0
wait until frame rate expired
Y
loopNum++
numBars=NUMBER_OF_SLOT?:1:++
loopNum = 1 ?
Y
SEND BLANK numBars = 1
N
Synch=0? OR loopNum < MAX_FRAMES
N
Y
numBars = assignedSlot?
N
SEND BLANK
Y
SEND 2 QRs sequentially based on ID & 2*numBars-1 2*numBars
loopNum > WAIT_FOR_ALL_PICO_SENDING_QR AND synch = 0 ?
N
Y
thread1 screenShot Created?
N
Create thread1 screenShot
Y
shotAnalyzed?
N
Y
synch = 1
locked = 1
END synchronization
START screenShot
locked?
Y
N
Pause FIFO Take screenshot to shotXX file
projectedID = valid ID?
N
Y
Get frame number values from all detected QR codes
determine the wait time for this Pico system = diff from the smallest
shotAnalyzed = Y
EXIT screenShot

D Tiled and Super Imposed Setting – Software Flow Chart


BEGIN
Initial parameters
Fading parameter calculation (tiled only)
Increase Loop number
Loop number > MAX_FRAME?
Y
N
Setup Load movie
Update pixels
get pixels
fading calculation (tiled only)
update pixels to blending (tited only)
Homography transformation
Draw graphic to display output
Y
Loop number > CAPTURE_START ?
N
SYNC_FLAG=ON?
Y
Time2wait > 0 ?
N
N
Y
Increase number of trials
Decrease Loop number by Time2wait and clear Time2wait
# trials > MAX_TRIALS?
Y
Increase one time step
reset number of trials
Project QRs
wait for frame time expired
Capture Frames
Save captured frame to file
load captured file
Decode QRs (long time)
Detected valid QRs
N
Y
QR code saved to buffers
Clear QRs buffers
QR corners saved to file
Frame ID saved to buffers
All QRs received?
N
Y
Load QR corner file
Calculate Homography matrix
Save calculated Homography matrix to file
Calculate delay frame time to wait
SYNC_FLAG=ON
EXIT