Rectangular Stable Power-Aware Mobile Projection on Planar Surfaces

Mehdi Rahimzadeh[*] UC Irvine
Hung Nguyen[†] UC Irvine
Ardalan Amiri Sani[‡] UC Irvine
Fadi Kurdahi[§] UC Irvine
Aditi Majumder[¶] UC Irvine

Abstract

Pico projectors are becoming increasingly popular and can be used in conjunction with handheld mobile devices, portable media players, digital cameras in the foreseeable future. Mobile projection provides new opportunities realizing mobile spatial augmented reality applications. Projection from handheld devices bring in challenges of stabilizing the imagery in the persence of hand movements which entails more complex problems like keystone, zoom and jitter correction. However, no suitable software interface exists today that integrates these pico-projectors with handheld mobile devices.

This paper develops the first power-efficient software methods and interface to project a rectangular, stable image from a pico-projector augmented to a handheld mobile device that can allow users to make contents legible on such devices via real-time solutions for keystone correction, zoom and jitter correction. To provide power efficient solutions in the context of mobile power-sensitive devices, we achieve the keystone correction, zoom and hand-movement stabilization mostly using data from an inertial measurement unit (IMU) while minimizing the use of less power-efficient sensor like camera. For zoom stabilization, we use a very inexpensive low power distance sensor. The camera is used only for initialization. We have developed two prototype systems to show our software interface: the first one is implemented on cellphone running Android OS; and the second one used a raspberry pi interface to the projectors.

Keywords: Keystone correction, Projector stabilization, projector calibration

Concepts: •**Computing methodologies** → *Mixed / augmented reality; Image processing;*

1 Introduction

Digital projectors have been used for presentation on big screens by setting them on a table or mounting them off the ceiling projecting perpendicular to the screen. But, recently, we have seen advancement of much smaller projectors — often called a pico or pocket projector. Emergence of these small and light-weight projectors along with the widespread use of smartphones opens up the opportunity to augment smartphones with projectors thereby enabling a large number of spatially augmented reality applications where physical objects are augmented with digital contents. Examples include appearance editing, huddle-free group experiences, gaming and so on. The same methods can also be used to mount projectors on unmanned aerial vehicles (UAV) to provide a remotely controlled or automated light output mechanism on such devices.

Mobile augmented reality applications continuously use mobile camera to augment virtual object on a image of real scene. In spatial augmented reality a projector is needed to project the image of virtual object on the physical scene. Both camera and projector are power-hungry devices, and using them simultaneously drains the mobile battery very fast. Therefore, using IMU instead of camera for correcting and stabilizing the projected imagery is useful.

Unlike panel-based displays, projectors bring in a tremendous flexibility to project on regular planar surfaces like walls, floors and ceilings and at a much larger size (10”-16” diagonal) than is possible from current mobile displays enabling huddle-free multi-user viewing. However, projecting on regular surfaces bring in challenges of correcting keystone distortions, stabilizing imagery in the presence of hand movements, adjusting zoom with movement of projectors towards and away from the surface. There has been several prior works on solutions to these problems when using single or multi-projector displays for presentation or virtual reality environments using cameras [Raskar and Beardsley 2001; Sukthankar et al. 2001; Li et al. 2011; Li and Sezan 2004]. However, the unique challenge in a mobile environment is its power sensitivity. Traditional camera based solutions would drain power very fast making existing methods non-usable in such environments.

1.1 Main Contributions

In this paper we present power-aware software methods and interfaces to achieve a rectangular shaped stable projection from an hand-held mobile device. Instead of cameras we use other available or easily augmentable sensors in cell phones like an IMU or a distance sensor. The camera is used only minimally when it is absolutely critical. To the best of our knowledge, this is the first work that addresses rectangular, stable and power-aware projection from mobile devices on flat or close-to-flat objects thereby enabling their use on mobile devices or UAVs as light output devices.

In particular, following are our technical contributions.

1. When projected on a flat surface, the projection usually looks trapezoidal and not rectangular. This arises due to the perspective projection when the projection axis is not perpendicular to the surface we are projecting on. This is more commonly referred to as *keystone correction.* We present a method to correct for keystoning using only IMU thereby resulting in a rectangular projection.

2. The mobile projectors come in different technology. While some use laser projectors, others use DMD arrays (digital micromirror arragys). The former can focus anywhere due to a scanning beam based technology (like CRT displays), while the latter can only focus at a preset distance from the device. Hand movements can also make the picture blurry due to deviation from the zone of preset zoom. We use the measurement from an inexpensive low-power distance sensor to maintain a static zoom setting during such movements.

[*] e-mail:rahimzam@uci.edu
[†] e-mail:hnguye17@uci.edu
[‡] e-mail:ardalan@uci.edu
[§] e-mail:kurdahi@uci.edu
[¶] e-mail:majumder@ics.uci.edu

VRCAI ’16,, December 03-04, 2016, Zhuhai, China
ISBN: 978-1-4503-4692-4/16/12...$15.00
DOI: http://dx.doi.org/10.1145/3013971.3013989

3. Projection from hand-held devices can show jitter due to small involuntary hand movements. The same can occur for devices on UAV which are known for their characteristic hovering movements. We present a method for correcting the jitter of the projected image in such situations.

4. Finally, for all the aforementioned methods, we need to know the intrinsic (focal length) and extrinsic parameter (position and orientation) of the projector. Therefore, we have developed an interface as part of the user application to calibrate his projector. Availability of such calibration process in the application provides flexibility for the user to augment different kinds of projector to their mobile phone. The camera is used only during this initializtaion of the system following which we resort back to our non-camera sensors to achieve stable, rectangular, imagery.

5. We demonstrate all the above capabilities using two prototypes: a cell phone connected to a pico-projector using Android OS; and a raspberry PI interface to the pico-porojector.

1.2 Related Work

Projection-Based Displays: In the computer graphics, vision and virtual reality community there has been a large amount of work on camera based automated solutions for registering one or more projectors on planar or non-planar surfaces using one or more cameras. A good survey of such works is available in [Brown et al. 2005; Majumder and Sajadi 2013]. In particular, [Raskar and Beardsley 2001; Sukthankar et al. 2001; Li et al. 2011; Li and Sezan 2004] uses camera-based keystone correction for a single presentation projector on a flat surface. These works does not assume continuous movement of the projector as would be the case for a hand-held device and therefore only achieves an one time correction during projector setup for presentation. Other works [Yang and Welch 2001; Cotting et al. 2004; Cotting et al. 2005; Johnson and Fuchs 2007] uses camera-based corrections (sometimes using two cameras with each projector) of single projectors in the face of continuous projector movements, but these are usually achieved on non-flat surfaces that provide enough visual features to constrain the problem for camera-based solutions. Flat surfaces are particularly hard for cameras to achieve any kind of projector calibration due to the lack of geometric features to constrain the problem sufficiently. Some recent works [Xu et al. 2013] therefore use depth camera to find the normal vector of the surface we are projecting on to correct for keystoning distortions.

However, all the aforementioned works are inapplicable in our scenario primarily because *camera based sensing does not provide a power-efficient solution.* Therefore, we use multiple inexpensive and power efficient sensors, like the Inertial Measurement Unit (IMU) and the distance sensor, to achieve the rectangular stable imagery in a power efficient manner. However, since these sensors provide different kinds of non-visual information than a camera, we need to develop different algorithms to achieve the rectangular stable imagery. Further, these devices are far more noisy than a camera. Therefore, we have to develop techniques that are resilient to such noisy input.

Camera Image Stabilization: The mode of operation of a camera is point-and-shoot where keystoning or zoom problems usually don't arise. However, correcting for jitter in hand-held cameras while capturing pictures has been an issue, especially for long exposure captures. There has been prior works in handling such jitter as a non-realtime postprocessing [Grundmann et al. 2012; Gleicher and Liu 2007; Matsushita et al. 2006]. Some recent works [Bell et al. 2014; Karpenko et al. 2011; Hanning et al. 2011] use IMU data in real-time to predict the movement of the camera window during video capture. We draw inspiration from these work to achieve jitter correction in projectors, however we need to achieve this in realtime.

2 Background

First, we briefly explain the device models, non-camera based sensors, and coordinate systems used in this paper.

2.1 The Projector Model

In this paper we use a linear camera and projector model, which are considered to be dual of each other with the direction of the optical path reversed [Faugeras 1993; Raskar et al. 1998]. The light from a 2D pixel (x, y) in the projector P (or a camera C) is related to the 3D surface point (X, Y, Z) it illuminates (or captures) via a linear 3×4 matrix M as

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = M \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{1}$$

where the 3D points and the 2D projections are expressed as 4D and 3D homogeneous coordinates respectively. The matrix M comprises of the 3×3 intrinsic parameter matrix K and a 3×4 extrinsic parameter matrix E. While the former does not change with the position and orientation of the projector, the latter does. If the position of the projector in 3D is given by $T = (x_t, y_t, z_t)^T$ and its orientation is given by a 3×3 rotation matrix R, then M is given by

$$M = KE = \begin{pmatrix} f & s & p \\ 0 & f & q \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{RT} \end{pmatrix}, \tag{2}$$

whose 2D location is given by $(x, y) = (\frac{X'}{Z'}, \frac{Y'}{Z'})$. Here, f is the focal length of the projector, p and q denote the location of the center of the image plane and s is a skew parameter which is nonzero only when the image coordinate axes are not orthogonal to each other.

We also use the well-known concept of homography that says if the surface illuminated by P is a planar, as is the focus of this paper, the relationship between the surface coordinates (u, v) and the projector coordinates (x, y) is related by an invertible 3×3 homography, H, given by

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \tag{3}$$

When using camera based sensor, the homography can be recovered using a minimum of four point correspondences between the projector coordinates (x, y) and the display coordinate (u, v).

2.2 Coordinate Systems

In this section, we define the different coordinate systems we use, illustrated in Figure 1. The first one is a *global coordinate system*, often called the static frame in contemporary literature. We denote this by C_g defined by the axes X_g, Y_g and Z_g. Here Z_g is reverse of gravity and toward the sky, and Y_g is toward the magnetic North. Therefore, X_g is toward the East.

The second coordinate system is the *device coordinate system* attached to the mobile device denoted by C_d defined by (X_d, Y_d, Z_d). Z_d is defined as normal to the panel display, Y_d is parallel to the

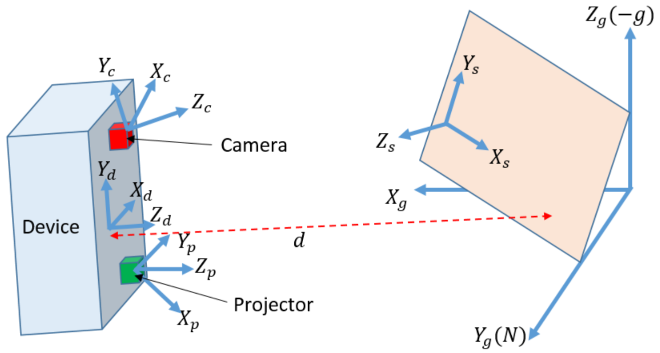

Figure 1: *Different coordinate systems used in this paper is illustrated. Projector C_p, camera C_c, and the device C_d coordinates are on the rigid body of the mobile device. The global C_g and planar surface coordinates C_s have fixed translation.*

panel display, and $X_d = Y_d \times Z_d$. Note that C_d is related to C_g via a translation and rotation denoted by $R_{d\to g}$ and $T_{d\to g}$ respectively.

The projector and camera connected to the device has their own coordinate system denoted by C_p and C_c respectively. But, since the projector and camera are rigidly integrated within the device, C_p and C_c are related to C_d by a rigid body transformation given by $T_{p\to d}$ and $T_{c\to d}$ respectively, *that does not change with movements of the device*.

The final coordinate system is that of the planar display on which the device is projecting, denoted by C_s. We assume Z_s to be normal to the flat display surface pointing towards the mobile device from which it is illuminated. The X_s and Y_s defines the plane of the surface.

2.3 Non-Camera Based Sensors

We use multiple non-camera sensors, the first one being the inertial measurment unit or IMU. It consists of a 3-axis gyroscope, a 3-axis accelerometer and a 3-axis magnetometer providing us with the measurements of angular velocity, linear acceleration, and magnetic field respectively. Ideally, the aforementioned non-camera based sensors can provide us with $T_{d\to g}$ and $R_{d\to g}$. However, all these devices provide noisy measurements – the signal to noise ratio being lower in the gyroscope than in the accelerometer. Therefore, it is well-known in the literature [Bell et al. 2014] that the $T_{d\to g}$ deciphered from this is very noisy and highly unreliable. Therefore, we use only the deciphered $R_{d\to g}$ from the IMU. However, we fuse the observations from the less noisy accelerometer measurements to aggregate the noisy measurements from the gyroscope using a Kalman filter based method presented in [Sabatini 2006] thus relating C_d and C_g. The column vectors of $R_{d\to g}$ provides the unit vectors along the coordinate axes of C_d.

The second non-camera sensor we use is an ultrasonic range sensor that provides the distance from the planar display, d, in centimeter. The measurement is accurate as long as the shortest distance between sensor and the surface is within the field of view of the sensor. This distance is used for zoom correction and projector initialization.

3 Problem Statement

When considering projection from hand-held mobile device on a static planar display, C_d changes with respect to C_s continuously. Our goal is to remove the distortions in the projected image due to these movements creating a stable imagery. We consider three different movements: change in orientation of the device due to rotation of the wrist, change in the zoom level of the device due to movement towards or away from the projection surface, and jitter in the device due to small rotation and translation in hand movements. Note that the effect of the first two movements is much more on the projected image than the last one. This is due to the fact that the image is formed on a distal surface where rotation and zoom creates a much larger movement or resizing of the image than the jitter. Let us now define the different corrections needed to achieve a rectangular stable projection from the mobile device.

Keystone Correction: When Z_p and Z_s is aligned (though opposite direction and therefore related by a scale factor of -1), the image looks rectangular on the flat display surface. However, when the orientation of the device changes, the display looks trapezoidal on the surface. This keystone distortion is compensated by estimating the homography between the surface and the projector coordinates which is then inverted to apply a warp to the projection image such that when projected image looks rectangular.

Zoom Correction: Even if the axes Z_p and Z_s are coincident, the size of the projected image will change as the device moves along Z_p. In order remove this zooming effect and maintain a standard size of the projection during such movements, we use a distance sensor to detect the distance of the planar surface from the mobile device and adjust the projected image size accordingly.

Jitter Correction: Finally, we apply a jitter correction inspired by [Bell et al. 2014] that compensates for the small rotations in the hand by computing the translation and rotation of the image it causes on the distal surface. Effects of small translations is close to imperceptible when compared o the effects of rotation and is therefore ignored.

4 Keystone Correction

In this section, we detail the theory of keystone correction. First, we detect orientation of the device via the rotataion angles between C_d and C_g from the non-camera sensor readings (Section 4.1). Next, we present the theory to relate these angles sensed by the non-camera sensors to the homography (Section 4.2). However, we find that this requires knowing the projector's intrinsic parameters that cannot be recovered just via non-camera sensors. Therefore, initially we use the camera once to initialize the homography, described in Section 4.3, following which the cameras are no longer used and the corrections are achieved completely using non-camera based sensors. *This helps us to use minimal power thereby making our method power-aware*.

4.1 Orientation from Non-Camera Sensors

In order to correct the keystone effect, we should know how the normal vector of the projector space (Z_p) deviates from the normal of the projection surface (Z_s). The IMU provides us with rotation between C_d and C_g. In order to find orientation of C_p with respect to C_s, we need to find the static rotation matrices for $R_{p\to d}$ and $R_{g\to s}$. The rotation matrix between C_p and C_s, $R_{p\to s}$, can be derived from these two as follows.

$$R_{p\to s} = R_{p\to d} R_{d\to g} R_{g\to s}. \tag{4}$$

To project an image that does not have keystone distortion Z_s should be aligned with $-Z_p$. To achieve this we need to apply two rotations to C_d, denoted by θ and ϕ, about X_p and Y_p respectively.

θ and ϕ are calculated akin to [Diebel 2006] as

$$(\theta, \phi) = \left(sin^{-1}(R_{p \to s}(3,1)), -tan^{-1}\left(\frac{R_{p \to s}(3,2)}{R_{p \to s}(3,3)} \right) \right). \quad (5)$$

These angles are are updated at every IMU readout and used for updating the homography matrix.

4.2 Relating Orientation to Homography

As the device orientation changes, the image plane of the projector gets tilted creating an oblique projection on the display surface. This is illustrated in a 2D depiction in Figure 2 where the perpendicular (undistorted) projection occurs for the orientation in red and the keystone projection occurs for the orientation in green when Z_p is no longer aligned with $-Z_s$. Note that in Figure 2 (Left), the distance of the device from the surface remains constant. The change in this distance is handled by zoom correction. Note that the center of projection of the tilted projector in green also changes. This is due to the fact that the rotation with movement is with respect to the coordinate axes C_d and not C_p.

Our goal is to make the image coming from the green titled projector in Figure 2 (Left) look like that coming from an imaginary red projector. This imaginary projector is derived by rotating the image plane of the green projector about its center making it parallel to the display surface. Therefore, the distance between the center of projection and the image plane of both these projectors should be identical. Let us denote the coordinate system of this imaginary projector as $C_p^{'}$. The homography matrix H in Equation 3 relates the 2D coordinates on the image plane of the red projector to that of the green projector.

Let us now consider the homography in Equation 3. Since the green and red projector's center of the image plane are coincident $h_{13} = h_{23} = 0$. Other entries of the homography matrix are categorized as projective entries (h_{11} and h_{22}), scaling entries (h_{11} and h_{22}), and the x-skew entry h_{12}. The projective entries are non-zero, because they represent the effect of oblique projection [Hartley and Zisserman 2003]. These parameters provide non-linear expansion and contraction of the content. This is because they construct the denominator in finding the image coordinates from the homogeneous coordinates. The scaling entries are non-zero because of the expansion of the content under oblique projection can be different in different directions. The x-skew term is non-zero because of the order of decomposing the rotation matrix into θ and then ϕ. However, the y-skew entry, h_{21} is zero and will be explained in details in Section 4.2.3. In the following sections we describe how the projective, scaling and skew entries of H are recovered.

4.2.1 Computing Projective Parameters

The projective terms, h_{31} and h_{32}, are dependent on the rotation θ and ϕ, and are due to the rotation of the projector's focal plane around X_p the Y_p respectively. To derive these parameters, let us first consider an identity homography matrix with only a non-zero term at h_{31}.

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ h_{31} & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (6)$$

Therefore, the image coordinates (x', y') is given by

$$(x', y') = (\frac{u}{w}, \frac{v}{w}) = (\frac{x}{h_{31}x + 1}, \frac{y}{h_{31}x + 1}) \quad (7)$$

Note that h_{31} causes a non-linear effect in the horizontal direction values and a linear scale in the vertical direction. First, we find the relationship between X_p and X'_p. Please refer to Figure 2 (Right) for this purpose.

The radius of the circle is the focal length of the projector f, O and O' are the center of projection of the green and red projector respectively. Therefore, a point on X_p, of the form $(x, 0)$, is mapped in X'_p to (x_1, z_1) given by $z_1 = x_1 \tan\phi$. From similar triangles we have:

$$\frac{x_0}{f} = \frac{x_1}{f + z_1} \Rightarrow x_0 = \frac{x_1}{1 + \frac{z_1}{f}} = \frac{x_1}{1 + x_1 . \frac{\tan\phi}{f}} \quad (8)$$

Comparing Equation 7 and 8 we find the projective term $h_{31} = \tan\phi / f$. Similarly h_{32} is calculated. Therefore, the projective terms of homography matrix are:

$$h_{31} = \frac{\tan\phi}{f}, h_{32} = \frac{\tan\theta}{f} \quad (9)$$

4.2.2 Computing Scaling Parameter

The projective parameters make sure that the pixels are not stretched after keystoning correction. However, to restore the size of the content, we need to compute the scaling entries of the homography matrix, h_{11} and h_{22}, that control the scaling in the x and y direction respectively. Our goal post projective and scale transformation is to map the point at (x_1, z_1) to $(x_0, 0)$, as shown in Figure 2 (Right). Following finding the projective parameters in Equation 9, the point $(x_0, 0)$ is mapped to $(x_1, 0)$. Now, we need to decipher the scaling parameters to map (x_1, z_1) to $(x_0, 0)$ that satisfy $\|(x_1, 0)\| = \|(x_1, z_1)\| cos\phi$ where $\|(x, y)\|$ denote the length of the vector (x, y). Similar calculations can be done to relate y to θ yielding scaling parameters of the homography matrix:

$$h_{11} = \cos\phi, h_{22} = \cos\theta. \quad (10)$$

4.2.3 Computing Skew Parameter

The projective and scaling parameters makes a trapezoidal projection due to keystoning into a parallelogram. But the skew parameter, h_{12} is then required to transform it to a rectangle.

Assume a simple solid rectangle centered at origin with $2L$ length and $2W$ width in C_p is transformed to the parallelogram. This can be done by a homography matrix as an identity matrix with the skew parameter as the only non-zero element. The top right corner of the rectangle, given by $a(L, W)$, is transformed to a' as

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} 1 & h_{12} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} L \\ W \\ 1 \end{pmatrix}. \quad (11)$$

Therefore, the coordinates of a' is given by

$$(x', y') = (\frac{u}{w}, \frac{v}{w}) = (\frac{L + h_{12}W}{1}, \frac{W}{1}) = (L + h_{12}W, W)$$

Note that the y coordinate is unaffected by the skew parameter. The effect is only on the x coordinate like a shear proportional to W. However, we need to consider the effects of the rotations by angles θ and ϕ on the skew parameter. Considering that, the final homography matrix for keystone correction related to the non-camera sensors is given by

$$H = \begin{pmatrix} cos\phi & cos\theta sin\theta sin\phi & 0 \\ 0 & cos\theta & 0 \\ \frac{\tan\phi}{f} & \frac{\tan\theta}{f} & 1 \end{pmatrix} \quad (12)$$

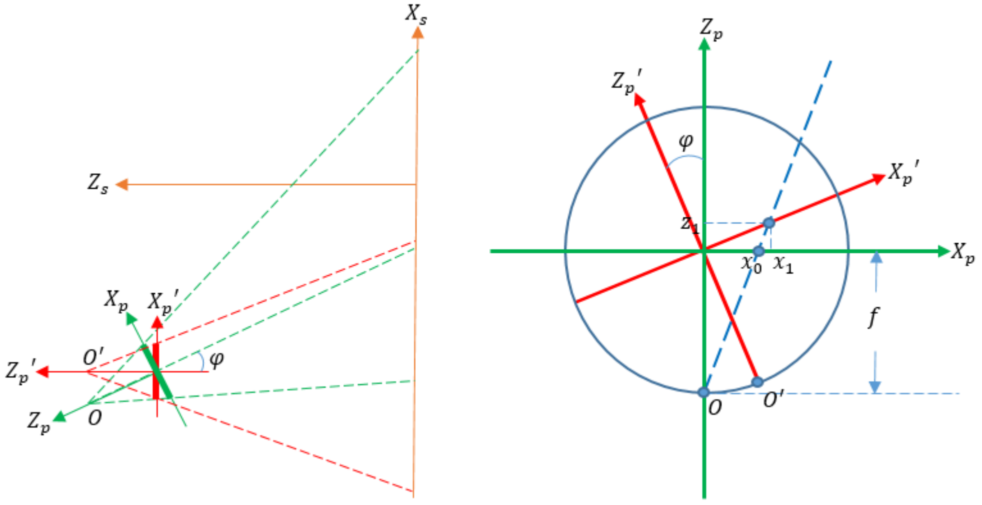

Figure 2: *Left: A 2D slice of projector and planar display is depicted. $X_p - Z_p$ of projector is shown in green, and $X_p^{'} - Z_p^{'}$, imaginary coordinate of the projector, is red. The imaginary coordinate is rotated to be parallel to $X_s - Z_s$, the coordinate of the planar display. Right: A mapping from $X_p^{'}$ to X_p based on the amount of rotation ϕ in closer view is depicted.*

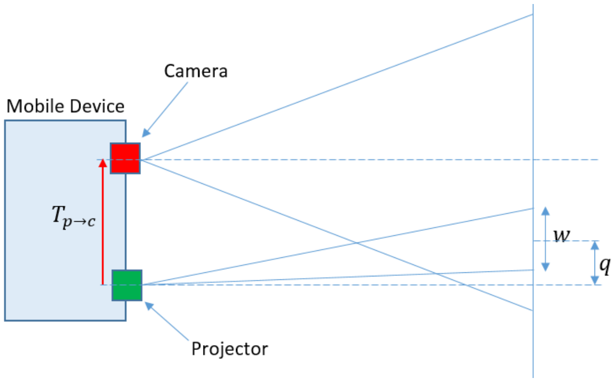

Figure 3: *This Figure shows the configuration of the projector and camera for initialization. They are separated by a fixed translation $T_{p \to c}$. The camera captures an image when the projector projects whole white image with width w. We can then find the projector's offset q.*

4.3 Initialization Using Camera

The homography defined with respect to parameters measured by non-camera sensors, presented in Equation 12, has a dependency on the intrinsic projector parameter, its focal length. This parameter can only be recovered using a camera. However, since it does not change dynamically, we can recover the homography matrix using a camera once in the very beginning. Following this initialization, we can update the homography matrix as the device moves only using the measured non-camera parameters. For this camera based initialization step, we make the following key assumptions.

First, we assume a calibrated camera. We apply the standard one time camera calibration method described in [Zhang 2000] that uses multiple images of a checkerboard to find the intrinsic parameters of the camera. To perform this process, we put a known checkerboard on the table and take a short video of the checkerboard while moving and rotating the mobile device on top of it.

Second, we assume that the transformation between projector (C_p), camera (C_c), and device (C_d) is known. Like other previous works [Karpenko et al. 2011; Bell et al. 2014], we also observe empirically that in most existing mobile devices there is no rotation between C_d and C_c. Only their z-direction is opposite achieved by a z-scaling of -1. Therefore, the transformation between C_d and C_c is reduced to a 2D translation, which is acquired from the device data sheet. Though no mobile device with projectors are commercially available today, we assume that the relationship between C_d and C_p are similar with no rotation and only a translation between them. We believe that such an assumption is reasonable given our prior experience with the Samsung projector enabled mobile device that was briefly introduced in the market around 2009. We assume that the translation between the camera and the projector is derived from the specification as $T_{p \to c}$. We also assume that the field of view (FOV) of the camera is larger than that of the projector as in [Bhasker et al. 2006]. Assuming that the non-malicious user who do not deliberately want to project a very key-stoned image, this assumption says that the image projected by the projector is within the camera FOV.

We denote the camera intrinsic parameter with K_C defined as

$$K_C = \begin{pmatrix} f_C & 0 & 0 \\ 0 & f_C & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad (13)$$

where f_C is the camera's focal length. We assume the principal center to be at the origin and the skew parameter to be 0. This assumption is true in practice in most low-cost cameras [Snavely et al. 2006]

We assume K is the intrinsic parameter matrix of the projector in Equation 2. Like a camera, it also depends on its focal length f. However, unlike cameras, the principal center of commodity projectors do not pass through the origin to accommodate offset projection [Sajadi and Majumder 2010]. Therefore, it has an offset in the y-direction, denoted by q. Therefore, the projector's intrisic matrix K is given by

$$K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & q \\ 0 & 0 & 1 \end{pmatrix} \quad (14)$$

Finally, we assume that the distance vector between projector and camera, denoted by $T_{p\rightarrow c}$ in known. The distance between the device and display d is measured from the proximity sensor also.

4.3.1 Computing Focal Length

Using all the aforementioned assumptions, we now detail our camera based initial projector calibration method. The user stands in front of a wall or floor or any other planar surface on which he wishes to project. The mobile device is held in such a way that Z_d axis is perpendicular to the planar display and projects a full-size all-white image. An image I of the projected image is captured by the camera that is processed in the following manner to calculate the projector's intrinsic parameters.

Note that the white rectangle projected by the projector can appear as a trapezoid in I. Therefore, we create an image I' from I that predicts the image seen by a virtual camera which is the dual of the projector. This is achieved by warping the image I using the fundamental matrix F as follows.

$$I' = K_C [T_{p\rightarrow c}]_\times K_C^{-1} I \quad (15)$$

where $[T_{p\rightarrow c}]_\times$ is the skew symmetric matrix of vector $T_{p\rightarrow c}$. This image will preserve the rectangular shape of the white projected rectangle. By using traditional image processing techniques, we compute the width of the projected rectangle as w pixels in image I'. From w, f_C and d, we find the size of the rectangle in centimeters, W, as

$$W = \frac{wd}{f_C} \quad (16)$$

Assuming w_p as the vertical resolution of the projector, the projector's focal length f in pixels is given by

$$f = \frac{dw_p}{W} \quad (17)$$

4.3.2 Computing Offset

Note that the center I' should be the center of the white rectangle if offset $q = 0$. However, due to a non-zero q, the center of the projected rectangle is above this central point. Let the distance of this point be Q pixels in image I'. The offset q is given from Q as

$$q = \frac{wQ}{w_p} \quad (18)$$

where w/w_p is ratio of the pixel pitch between the camera and the projector. Therefore, the final projector intrinsic parameter matrix is given by:

$$K = \begin{pmatrix} \frac{w_p f_C}{w} & 0 & 0 \\ 0 & \frac{w_p f_C}{w} & \frac{wQ}{w_p} \\ 0 & 0 & 1 \end{pmatrix} \quad (19)$$

4.4 Applying Keystone Correction

In realtime, we measure the angles θ, ϕ dynamically. Each measurement and the intrinsic matrix K is used to derive a matrix that is applied to the image to projected in real time to achieve dynamic correction resulting in a rectangular image even when the device orientation changes continuously. This is achieved by first translating the image by $(0, -q)$ to undo the effect of projector offset, followed by multiplication by the homography H, followed by the inverse translation by $(0, q)$. So, the keystone correction matrix is given by

$$T(0, q)\ H(\theta, \phi, f)\ T(0, -q) \quad (20)$$

where T denotes a translation matrix with parameters included in the parentheses.

5 Zoom Correction

The goal for zoom correction is to keep the projected imagery static as the projector moves towards and away from the planar display along Z_d. For this, we use the distance d measured by the distance sensor. First, we fix a reference distance (d_r) based on initial reading from the sensor to which the zoom factor is locked in. Small deviations of d from d_r is attributed to involuntary movements on the part of the user and therefore, the image size is not changed. Only when a large change happens, we reset d_r assuming that the user intends to change the zoom factor.

The relation between the projector and planar display coordinates where $-Z_p$ and Z_s are aligned after keystone correction is:

$$x_p = \frac{f}{d} x_s \quad (21)$$

where x_p and x_s are the value of the x-coordinates in C_p and C_s respectively. In order to correct for the changing zoom, we keep x_s static by changing x_p as

$$\frac{x_p^{(2)}}{x_p^{(1)}} = \frac{d_1}{d_2} \quad (22)$$

where $x_p^{(1)}$ and $x_p^{(2)}$ denotes the input and output image coordinates, respectively. Since this ratio can be considered as the scaling factor, the zoom correction is modeled as a 2D scaling matrix used to warp the content as

$$\begin{pmatrix} x_p^{(2)} \\ y_p^{(2)} \end{pmatrix} = \begin{pmatrix} \frac{d_r}{d} & 0 \\ 0 & \frac{d_r}{d} \end{pmatrix} \begin{pmatrix} x_p^{(1)} \\ y_p^{(1)} \end{pmatrix} \quad (23)$$

6 Jitter Correction

Jitter refers to small rotation and translations of the hand during projection that causes jittering of the projected imagery which is unpleasant for the viewer. As mentioned in Section 2.3, translation deciphered from the noisy IMU accelerometer measurements is not robust in the face of a double integration [Karpenko et al. 2011]. However, [Bell et al. 2014] faces the same problem in the context of camera image stabilization and shows that the jitter in the imagery caused by translation is perceptually negligible when compared to that of rotation, especially for large d. Therefore, we compensate only for the jitter in the rotation of the device. [Bell et al. 2014] also shows that due to the fixed translation between the IMU and projector, denoted by the matrix $T_{p\rightarrow d}$, small perturbation in rotation of the device should be modeled as a translation and rotation in projector. We use the same model for our jitter correction.

Equation 4 shows that small changes in $R_{d\rightarrow g}$ leads to small changes in the orientation of the projector $R_{p\rightarrow s}$. Therefore, as in zoom correction, we fix a reference rotation R_r based on the

IMU reading right after the keystone correction. At any subsequent step, let the rotation deciphered from IMU measurements be $R^{'}_{p\to s}$. Therefore, an additional rotation R_j due to jitter is computed as

$$R_j = R^{'}_{p\to s} \times R^{-1}_{ref} \tag{24}$$

The small translation in projector caused by R_j is given by the matrix T

$$T = R_j T_{p\to d} - T_{p\to d}. \tag{25}$$

The translation vector t is then extracted from the last column of T. In order to achieve a projected imagery that appears to be pasted to the surface despite jitter in the device, we apply the homography H_j given by

$$H_j = K[t]_\times R_j K^{-1} \tag{26}$$

where $[t]_\times$ is the skew symmetric matrix for the vector t.

7 Implementation and Results

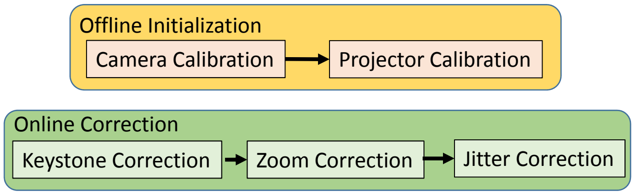

Figure 4: *The outline of the algorithm described in this paper. It is comprised of offline initialization and online correction parts.*

The final pipeline of our method consists of an off-line camera and projector calibration. Following that, during dynamic movements the keystone, zoom and jitter corrections are applied creating a stable, rectangular projection (Figure 4).

We created a prototype system comprising of a Nexus 5 smartphone from LG Electronics and a Pico Projector from MicroVision (Figure 5). These are attached in a manner such that the back camera of the phone and the projector are parallel with no rotation between them. The content from the panel display is sent to the projector via a SlimPort cable that provides connectivity from the phone's Micro USB to a HDMI video input for the projector. Any content shown on the display is therefore transformed to the projector in real-time. This prototype mobile device is also equipped with an 6-DOF axis IMU from InvenSense (MPU 6515). The location of the IMU module on the board is estimated from the phone's teared down maps as in [Karpenko et al. 2011], and illustrated in Figure 5(Left). The baseline between the projector and camera is measured manually. We have used an Ultrasonic Ranging module HC-SR04 for capturing the projection distance, which is connected to a raspberry pi board. This sensor provides 2-400 (cm) with $\pm 3mm$. The FOV of this sensor is 30°.

Our algorithm (Figure 4) is developed as an Android application. We utilized OpenCV library for Android for various warping and other matrix based computations. This application for creating a stable and rectangular image is developed in targeted OS Android 6.1. This application has a menu (Figure 5)the user can select the calibration or projection mode. In Calibration mode the camera and projector will be calibrated. Camera calibration is done by taking a short video of a checkerboard. The projector calibration part instructs the user to stand at the desired distance from the wall, and the projector projects all white. Then the application takes the photo of the white projected imagery. The calibration data is saved on the device. If the user selects the online correction from the menu, it shows the image selected by the user and lets the user to select the projection surface. As the user moves the device while projecting on the planar surface, the algorithm automatically performs the keystone correction. Once the user is determined where to project, they can select jitter correction from the menu to activate the stabilization algorithm along the pipeline. When the user exits the application, it starts from the main menu. The zoom correction is also triggered to compensate involuntary movement toward and backward of the planar surface. Figure 6 (two Right images) shows some results. Please see the video for dynamic results of jitter correction.

8 Conclusions

In this paper we provide the first solution for equipping mobile devices with projectors. Mounting projectors as output devices on mobiles open up several new directions for many applications that need larger projections for multi-user viewing. Achieving this just using IMU data restricts power usage making this as a energy efficient option to communicate in the future. We mostly focused on solutions with the help of IMU, since they are becoming more and more available in mobile devices as they are becoming less expensive, getting higher precision, and consuming lower energy. But, there are other sensors such as depth sensor, location sensing systems that may be helpful. These are some of the future directions we plan to follow. In terms of application development, it is possible in Android to have multiple video output. Therefore, the projected content might be different from that of shown on the screen. This feature provides feasibility for some applications that need projection, while the buttons and control information can be displayed on the screen. That would be useful in developing some new kinds of interaction modalities using mobile devices (e.g. games and entertainment).

References

BELL, S., TROCCOLI, A., AND PULLI, K. 2014. A non-linear filter for gyroscope-based video stabilization. In *European Conference on Computer Vision*, Springer, 294–308.

BHASKER, E. S., SINHA, P., AND MAJUMDER, A. 2006. Asynchronous distributed calibration for scalable and reconfigurable multi-projector displays. *IEEE Transactions on Visualization and Computer Graphics 12*, 5, 1101–1108.

BROWN, M., MAJUMDER, A., AND YANG, R. 2005. Camera-based calibration techniques for seamless multiprojector displays. *IEEE Transactions on Visualization and Computer Graphics 11*, 2, 193–206.

COTTING, D., NAEF, M., GROSS, M., AND FUCHS, H. 2004. Embedding imperceptible patterns into projected images for simultaneous acquisition and display. 100–109.

COTTING, D., ZIEGLER, R., GROSS, M., AND FUCHS, H. 2005. Adaptive instant displays: Continuously calibrated projections using per-pixel light control. 705–714.

DIEBEL, J. 2006. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix 58*, 15-16, 1–35.

FAUGERAS, O. 1993. *Three-dimensional computer vision: a geometric viewpoint*. MIT press.

GLEICHER, M. L., AND LIU, F. 2007. Re-cinematography: improving the camera dynamics of casual video. In *Proceedings of the 15th ACM international conference on Multimedia*, ACM, 27–36.

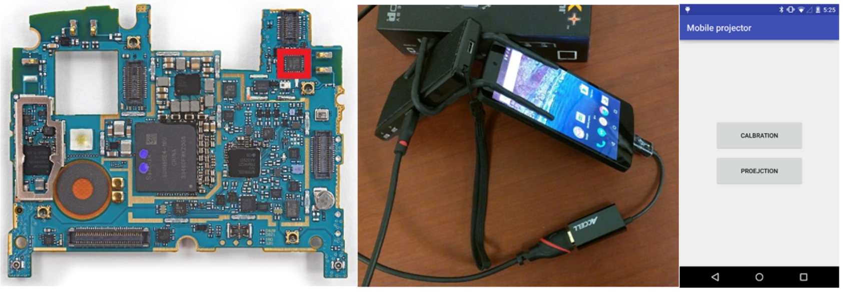

Figure 5: *Left: The teared-down Nexus 5 is shown and the IMU is annotated with red square (Courtesy: ifixit.com). Middle: It depicts our setup of Nexus 5 phone augmented with a pico projector. Right: The main page of the Android application is shown and user can choose to project an image or calibrate the system.*

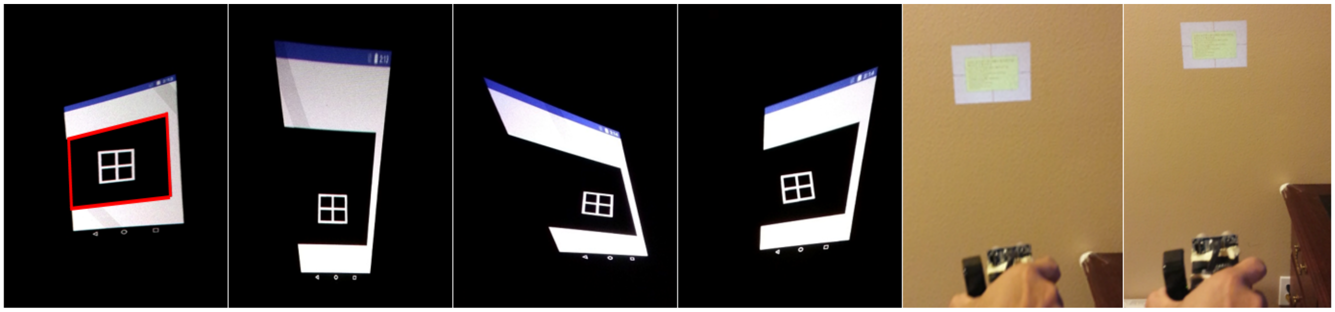

Figure 6: *The four Left images are the results from keystone correction under different values of θ and ϕ. There is a black trapezoid (annotated with red lines in the first image) inside the projected mobile screen, and a square window-like image inside that black box. The window-like image is warped with keystone correction homogoraphy. The keystone correction is visible if you compare the window-like image with its black box boundaries. In the first two image the device (and projector) are rotated around Y_p and X_p, respectively. In the next two images, the device is rotated in both X_p and Y_p. The two Right images show the zoom correction As the projector module, equipped with distance sensor, moves toward and away from the screen, the zoom correction factor is applied to keep the imagery size constant.*

GRUNDMANN, M., KWATRA, V., AND ESSA, I. 2011. Auto-directed video stabilization with robust l1 optimal camera paths. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, 225–232.

GRUNDMANN, M., KWATRA, V., CASTRO, D., AND ESSA, I. 2012. Calibration-free rolling shutter removal. In *Computational Photography (ICCP), 2012 IEEE International Conference on*, IEEE, 1–8.

HANNING, G., FORSLÖW, N., FORSSÉN, P.-E., RINGABY, E., TÖRNQVIST, D., AND CALLMER, J. 2011. Stabilizing cell phone video using inertial measurement sensors. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, IEEE, 1–8.

HARTLEY, R., AND ZISSERMAN, A. 2003. *Multiple view geometry in computer vision*. Cambridge university press.

HENG, L., LEE, G. H., AND POLLEFEYS, M. 2015. Self-calibration and visual slam with a multi-camera system on a micro aerial vehicle. *Autonomous Robots 39*, 3, 259–277.

JOHNSON, T., AND FUCHS, H. 2007. Real-time projector tracking on complex geometry using ordinary imagery. 1–8.

KARPENKO, A., JACOBS, D., BAEK, J., AND LEVOY, M. 2011. Digital video stabilization and rolling shutter correction using gyroscopes. *CSTR 1*, 2.

KUIPERS, J. B., ET AL. 1999. *Quaternions and rotation sequences*, vol. 66. Princeton university press Princeton.

LI, B., AND SEZAN, I. 2004. Automatic keystone correction for smart projectors with embedded camera. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, vol. 4, IEEE, 2829–2832.

LI, Z., WONG, K.-H., GONG, Y., AND CHANG, M.-Y. 2011. An effective method for movable projector keystone correction. *IEEE Transactions on Multimedia 13*, 1, 155–160.

LIU, S., YUAN, L., TAN, P., AND SUN, J. 2013. Bundled camera paths for video stabilization. *ACM Transactions on Graphics (TOG) 32*, 4, 78.

MA, Y., SOATTO, S., KOSECKA, J., AND SASTRY, S. S. 2012. *An invitation to 3-d vision: from images to geometric models*, vol. 26. Springer Science & Business Media.

MAJUMDER, A., AND SAJADI, B. 2013. Large area displays: The changing face of visualization. *Computer*, 5, 26–33.

MATSUSHITA, Y., OFEK, E., GE, W., TANG, X., AND SHUM, H.-Y. 2006. Full-frame video stabilization with motion inpainting. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 7, 1150–1163.

RASKAR, R., AND BEARDSLEY, P. 2001. A self-correcting projector. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2, IEEE, II–504.

RASKAR, R., WELCH, G., CUTTS, M., LAKE, A., STESIN, L., AND FUCHS, H. 1998. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, 179–188.

SABATINI, A. M. 2006. Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing. *IEEE Transactions on Biomedical Engineering 53*, 7, 1346–1356.

SAJADI, B., AND MAJUMDER, A. 2010. Auto-calibration of cylindrical multi-projector systems. In *2010 IEEE Virtual Reality Conference (VR)*, IEEE, 155–162.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, vol. 25, ACM, 835–846.

SUKTHANKAR, R., STOCKTON, R. G., AND MULLIN, M. D. 2001. Smarter presentations: Exploiting homography in camera-projector systems. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, IEEE, 247–253.

XU, W., WANG, Y., LIU, Y., WENG, D., TAN, M., AND SALZMANN, M. 2013. Real-time keystone correction for hand-held projectors with an rgbd camera. In *2013 IEEE International Conference on Image Processing*, IEEE, 3142–3146.

YANG, R., AND WELCH, G. 2001. Automatic projector display surface estimation using every-day imagery; presented by herman towles at the 9th international conference in central europe on computer graphics, visualization and computer vision 2001. *Plzen, Czech Republic*.

ZHANG, Z. 2000. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence 22*, 11, 1330–1334.