An Effective Method for Movable Projector Keystone Correction
IEEE Transactions on Multimedia, VOL. 13, NO. 1, Feb 2011. (Draft)

A Method for Movable Projector Keystone Correction

Zhaorong Li[1] *, Kin-Hong Wong[1], Yibo Gong[1], and Ming-Yuen Chang[2]
Dept. of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong[1]
Dept. of Information Engineering, The Chinese University of Hong Kong, Hong Kong[2]
E-mail: {zrli,khwong,ybgong}@cse.cuhk.edu.hk, {mchang}@ie.cuhk.edu.hk

***Abstract*—In this paper, we propose a hand-held mobile projection system that can freely project keystone free content on a general flat surface without any markings. Such a projection system can give the user great freedom of control of the display such as viewing angle and distance without suffering from distortion. We attach a camera to the projector to form a stereo pair. The correction of the display content is then achieved by rectifying the projection region of interest into a rectangular region and pre-warping the display content. Experimental results show that our system can continuously project distortion free content in real time with reasonable precision.**

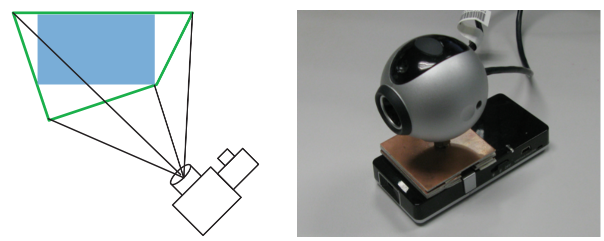
***Index Terms*—Projector, mobile projection, calibration, keystone correction**

I. Introduction

As technique advances, the size of projectors is shrinking quickly, which incubates the appearance of mobile projectors or mobile devices with embedded projector. These mobile projection devices provide us with highly enhanced viewing experience, through which our eyesight will no longer be limited in a small screen, nor will it be confined within a narrow angle. For example, using a digital camera with a projection module on it, we can shoot a picture and immediately project it in front of us to share with our friends, without having to ask them to stare at the small screen on the camera.

The promising future of mobile projection is so obvious. However, a big obstacle of its being widely used is an inherited limitation of projectors known as keystone distortion: when we project an image onto a screen at oblique positions, the projection region will become a trapezoid instead of a rectangle. This kind of distortion gives the user an unpleasant experience and the correction of it becomes a stringent need, especially in a mobile scenario where the mobile projector may be moving continuously. In this scenario, a good keystone correction method should be equipped with the following features: first, screen independent; no specially designed or position-fixed screen should be required, i.e., the user can project on any normal flat surface; second, continuous processing in real time; since the pose of the projector is not fixed, continuous correction instead of one-time correction is expected to be performed in real-time for the best of user experience.

Motivated by this, we propose a method which can continuously correct the distortion and display the content of interest in a rectangular area on a markless screen. The only additional device used is a webcam attached with the projector (see Fig. 1), which is quite natural since we are observing more and more mobile devices with both embedded projector and camera recently. Our method first recovers the 3D projection region. After that, we look for an inscribed rectangle in it and pre-warp the original display image so that it will be projected into this rectangle. While the projector is moving, the system keeps detecting the projection region and warping the projection content. As a result, the user can enjoy a keystone free viewing experience no matter how he or she moves the projector.



(a) (b)

Fig. 1. (a) Hand-held projector keystone correction. (b) The mobile projector attached with a webcam in our prototype.

Our method is specially designed for markless mobile projection, which is different from existing approaches concentrating on one-time correction for static projectors. For example, Sukthankar *et al.* [1] proposed to correct keystone with a fixed camera-projector pair by using homographies among the projector, camera and screen. However, in their proposed method, a fixed screen is needed, and the correction algorithm relies on detecting the screen boundary. This method is not suitable for mobile projection since blank surfaces (walls, floors etc) without boundaries or markings are usually used as the projection surface of mobile projectors. Raskar *et al.* [2] proposed a correction method without requiring boundary or markings on the screen. However, their algorithm requires a full calibration and use of the intrinsic and extrinsic parameters of the projector and camera. The complexity and high computation cost of the algorithm prevents it from being widely used in mobile projection applications. Li *et al.* [3] proposed an efficient keystone correction method in which not only keystone correction but also auto zooming and screen fitting are achieved. However, this method still requires a bounded screen. To sum up, existing methods are not suitable for our markless mobile projection purpose.

The remainder of the paper is organized as follows. In Section II, we give an overview of the proposed method.

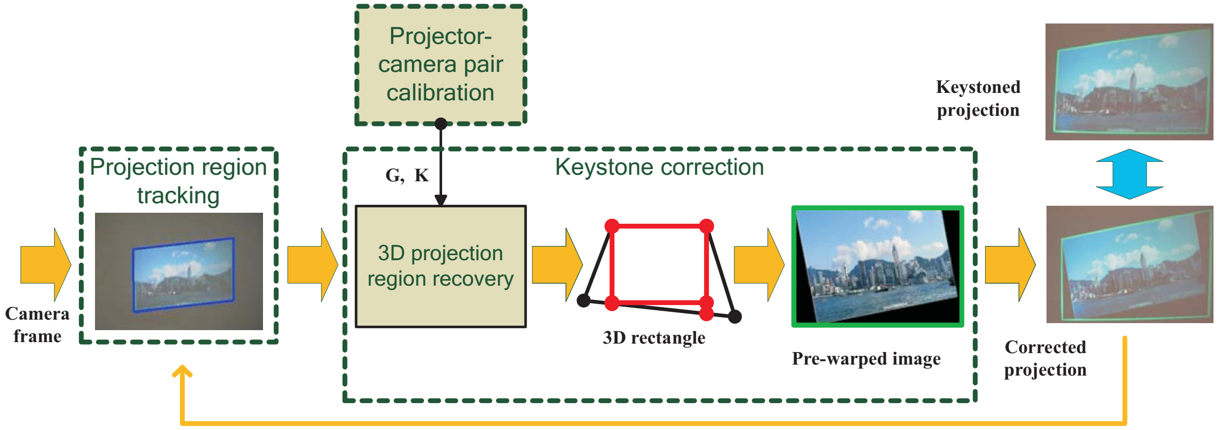


Fig. 2. The keystone correction flowchart.

The technical details are described in Section III, IV, V. Experimental results are given in Section VI. We conclude the paper in Section VII.

II. System overview

Our method is an integration of three modules, the calibration module, the tracking module, and the correction module. The calibration module is an one-time module, which finds the relationship between the projector and camera. The tracking module takes the camera capture as input, and tracks the projection region in camera. Based on the calibration result and the tracked projection region, the correction module rectifies the keystone distortion. The work-flow of the system is shown in Fig. 2. In following sections, we describe each module one by one in detail.

III. Projector-camera pair calibration

We use a calibrated camera with known intrinsic parameter matrix. In order to correct the keystone distortion, we need to calibrate the geometric relationship between the projector and camera. Traditional calibration methods for static projector-camera system usually estimate a projector-camera homography to represent the relationship. However, it is not suitable for a mobile projection system since the projector may be moving in this kind of system. The changing relative position between the projector and screen makes the projector-camera homography change accordingly. Hence, we should calibrate a fixed relationship between the projector and camera which is independent from the motion of the projector. Our solution is to find the projection matrix from the camera coordinate to the projector image.

In ideal situation, the projective model of a projector is similar to the camera model except for the projection direction. The projection from a 3D world point to the 2D projector image pixel is also via a 3×4 perspective projection matrix. So for each 3D point $\mathbf{X}^{\mathrm{c}}(x, y, z)$ in the camera coordinate system, it relates its corresponding projector image pixel $\mathbf{x}^{\mathrm{p}}(u, v)$ by a projection matrix $\mathbf{G}$:

$$s\tilde{\mathbf{x}}^{\mathrm{p}} = \mathbf{G}\tilde{\mathbf{X}}^{\mathrm{c}} \quad (1)$$

where $\tilde{\mathbf{x}}^{\mathrm{p}}, \tilde{\mathbf{X}}^{\mathrm{c}}$ are homogeneous coordinates, s is a scale factor, $\mathbf{G}$ is the projection matrix describing the intrinsic parameters of the projector and the relative pose between the projector and camera:

$$\mathbf{G} = \begin{pmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \end{pmatrix} \quad (2)$$

An explicit calibration like [2] involves estimating all the intrinsic and pose parameters, which is complicated and not easy to obtain a stable result. However, in our system, owing to our novel keystone correction algorithm, we do not need to estimate all these parameters explicitly, but simply estimate the projection matrix $\mathbf{G}$.

A simple method proposed in [4] is employed to estimate the projection matrix, the main idea of which is to collect a number of correspondences between the projector and camera. An ordinary cardboard with known size is used as the calibration object. The user holds the cardboard and freely moves it in front of the camera. At the same time, a cross with known position is projected onto the cardboard. The calibration module automatically detects the cardboard and the cross in the camera. The 3D positions of the cardboard and the cross in the camera coordinate are then easily calculated via Zhang's method [5]. In this way, a 3D-2D correspondence is obtained. The projection matrix is then estimated based on a number of such 3D-2D correspondences using Singular Vector Decomposition (SVD).

IV. Projection region detection and tracking

To facilitate the detection of projection region, we add a green frame with full size of projector screen to the projection image. The whole detection process can be divided into two stages. In the initial stage, we detect a quadrangle fulfilling several criteria as the initial position of the projection region. After that, we track its position in the subsequent frames. The tracking process is introduced so as to obtain a smooth and coherent result.

A. Detection

The detection is performed on the edge map obtained by Canny edge detector. We use Hough Transform line detector to extract a set of line segments, and then test which four segments form a desired quadrangle with following criteria: (1) each side of the formed quadrangle should be longer than a threshold; (2) opposite sides should have similar lengths; (3) each angle should be within the range from $30°$ to $150°$; (4) the overlapping ratio of the line segments to the four sides of the formed quadrangle should be bigger than a threshold; (5) the quadrangle is located nearly in the center of the camera image. If a quadrangle satisfying all the criteria is detected, we regard it as the initial projection region and proceed to the tracking process.

B. Particle Filter tracking

From the detection result, we have obtained the 2D positions of the four corners of the projection region. A direct way is to track the four corners in the subsequent frames. However, there would incur redundancy in the tracking state since the projector camera pair is actually dominated by a homography:

$$s\tilde{\mathbf{x}}^{\mathrm{p}} = \mathbf{H}\tilde{\mathbf{x}}^{\mathrm{c}} \tag{3}$$

where $\mathbf{H}$ is the homography matrix from the camera to the projector, $\mathbf{x}^{\mathrm{p}}$ is the corner of the projector screen, $\mathbf{x}^{\mathrm{c}}$ is the corner of the projector region in camera. According to, $\mathbf{H}$ is further expressed as:

$$\mathbf{H} = \mathbf{J}[\mathbf{R} - \frac{\mathbf{t}\mathbf{n}^{\mathrm{T}}}{d}]\mathbf{K}^{-1} \tag{4}$$

where $\mathbf{J}$ is the intrinsic parameter matrix of the projector, $\mathbf{R}$ and $\mathbf{t}$ are the rotation and translation of the camera relative to the projector, $\mathbf{n}$ is the normal of the screen relative to the camera, d is the distance of the screen from the camera, $\mathbf{K}$ is the intrinsic parameter matrix of the camera. From Eq. (4), we can see that the projector-camera homography is actually ruled by $\mathbf{n}$ and d. So we can track $\mathbf{n}$ and d instead of four corners:

$$\mathbf{s} = [\mathbf{n}^{\mathrm{T}}, d] \tag{5}$$

The number of state parameter is accordingly reduced from 8 to 3.

1) dynamic model: Since the projector is moving in free motion, a simple random walk model based on an uniform density U about the the previous state is used. The variable e represents the uncertainty about the movement of the projector.

$$p(\mathbf{s}_k|\mathbf{s}_{k-1}) = U(\mathbf{s}_{k-1} - \mathbf{e}, \mathbf{s}_{k-1} + \mathbf{e}) \tag{6}$$

2) observation model: To evaluate the likelihood of each particle, we first re-project the sphere to the camera image plane according to the $[\mathbf{n}^{\mathrm{T}}, d]$ represented by the particle. The re-projection is done according to Eq. (4). Though we do not calibrate the projector parameters explicitly, i.e, we find $\mathbf{G}$ instead of $\mathbf{J}$, $\mathbf{R}$, $\mathbf{t}$, it is still feasible the re-projection. We reformulate Eq. (4) to Eq. (7):

$$\mathbf{H} = [\mathbf{J}\mathbf{R} - \frac{\mathbf{J}\mathbf{t}\mathbf{n}^{\mathrm{T}}}{d}]\mathbf{K}^{-1} \tag{7}$$

Then, $\mathbf{J}\mathbf{R}$ and $\mathbf{J}\mathbf{t}$ can be obtained from $\mathbf{G}$:

$$[\mathbf{J}\mathbf{R}, \mathbf{J}\mathbf{t}] \propto [\mathbf{G}_{3\times3}, \mathbf{G}_{3\times1}] \tag{8}$$

where $\mathbf{G}_{3\times3}$ means the first three columns of $\mathbf{G}$, $\mathbf{G}_{3\times1}$ means the last column of $\mathbf{G}$. In this way, we can obtain a homography matrix. The the projector screen is then back-projected to the camera image by the inverse homography matrix:

$$s\tilde{\mathbf{x}}^{\mathrm{c}} = \mathbf{H}^{-1}\tilde{\mathbf{x}}^{\mathrm{p}} \tag{9}$$

After re-projecting the quadrangle, we evaluate the particle's likelihood of being the desired one by checking how many edge points are on the four sides of the quadrangle. The checking is performed along each side for every 5 pixels. If there is an edge point whose perpendicular distance to the side is within 5 pixels, we consider that the side has an on-edge point. The likelihood of that side is then assigned as the proportion of on-edge points among total points on that side, and the likelihood of that quadrangle is the sum of the likelihoods of all four sides. After all candidate quadrangles are evaluated, we choose the candidate position with the maximum likelihood as the final result of current frame.

3) Initialization: The detected quadrangle is used to initialize the particle filter. We reconstruct its 3D position in the camera coordinate system using the method in Section V-A. Its normal and distance to the camera are calculated accordingly. They are used as the initial state of the particle filter.

The proposed tracking algorithm is working well owing to the fact that the relationship between the projector and camera are fixed, which makes the projection region appear nearly rectangular in the camera; moreover, its movement is limited to several routes (translating, scaling, and skewing) within a small range.

V. AUTOMATIC KEYSTONE CORRECTION

The correction algorithm contains four steps. As shown in Fig. 2, we first detect the full screen projection region of the projector in the camera image. Second, the 3D position of the projection region is recovered based on the 2D detection result. Third, we look for an inscribed rectangle inside the 3D projection region. Finally, the original projection image is pre-warped so that it will be projected into the inscribed rectangle on the screen. This process repeats for each frame of the camera video during the movement of the projector.

A. Recovering 3D projection region

Having obtained the 2D camera position of the projection region, we proceed to recover its 3D position in the camera coordinate system. Based on the calibration result of the projector-camera pair, for each corner of the projection region, we can recover its 3D location from its corresponding pixels in the camera and projector image. In the projection image, the four corners are easy to obtain according to its resolution, and we have observed that no matter how we move the projector-camera pair, the orientation of the projection area viewed by the camera does not change. So the correspondence between the quadrangle corner in camera and projector is easy to establish.

Supposing that the 3D coordinates of the four corners of the projection region to be solved are $\mathbf{X}_i^c, i = 1\ldots4$ respectively, they and their 2D positions in the camera $\mathbf{x}_i^c(u,v)$ and projector images $\mathbf{x}_i^p(\alpha,\beta)$ should satisfy Eq. (1) and the projection equations of the camera in Eq. (10):

$$\lambda\tilde{\mathbf{x}}^c = \mathbf{K}\mathbf{X}^c \quad (10)$$

$$\mathbf{K} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \quad (11)$$

where λ is a scale factor, and $\mathbf{K}$ is the intrinsic parameter matrix of the camera. Each projection equation can be rearranged into two linear equations. Hence, there are totally 4 linear equations with 3 unknowns (3D coordinate of the corner). A least square solution can be obtained by SVD.

However, the SVD solution cannot guarantee the coplanarity of the four corners since they are solved separately. Geometrically, four coplanar points should satisfy the following condition:

$$\overrightarrow{\mathbf{X}_1^c\mathbf{X}_4^c} \cdot (\overrightarrow{\mathbf{X}_1^c\mathbf{X}_2^c} \otimes \overrightarrow{\mathbf{X}_1^c\mathbf{X}_3^c}) = 0 \quad (12)$$

Directly incorporating the condition into Eq. (1) and Eq. (10) will result in a nonlinear equation that is difficult to solve. Instead, we carry out a post refinement to the SVD solution, which minimizes the sum of back-projection errors in the camera and projector, plus the coplanarity constraint:

$$\begin{aligned} & \sum_{i=1}^{4} \|u_i - \frac{\mathbf{k}_1^T\mathbf{X}_i^c}{\mathbf{k}_3^T\mathbf{X}_i^c}\|^2 + \|v_i - \frac{\mathbf{k}_2^T\mathbf{X}_i^c}{\mathbf{k}_3^T\mathbf{X}_i^c}\|^2 \\ + & \sum_{i=1}^{4} \|\alpha_i - \frac{\mathbf{g}_1^T\tilde{\mathbf{X}}_i^c}{\mathbf{g}_3^T\tilde{\mathbf{X}}_i^c}\|^2 + \|\beta_i - \frac{\mathbf{g}_2^T\tilde{\mathbf{X}}_i^c}{\mathbf{g}_3^T\tilde{\mathbf{X}}_i^c}\|^2 \\ + & \omega\|\overrightarrow{\mathbf{X}_1^c\mathbf{X}_4^c} \cdot (\overrightarrow{\mathbf{X}_1^c\mathbf{X}_2^c} \otimes \overrightarrow{\mathbf{X}_1^c\mathbf{X}_3^c})\|^2 \end{aligned} \quad (13)$$

where $\mathbf{k}_1^T, \mathbf{k}_2^T, \mathbf{k}_3^T$ are three row vectors of $\mathbf{K}$, $\mathbf{g}_1^T, \mathbf{g}_2^T, \mathbf{g}_3^T$ are three row vectors of $\mathbf{G}$, ω is a weight. Taking the SVD solution as initialization, we use the Levenberg-Marquardt algorithm [6] to minimize the above function. The optimization stops if a pre-defined accuracy of coplanarity is reached. Since the SVD solution is already close to coplanarity, the above optimization stops within a few iterations.

B. Looking for inscribed rectangle

Next, according to the obtained 3D positions of the quadrangle's corners, we look for an inscribed rectangle inside the quadrangle whose top side resides in that of the quadrangle. Then the inscribed rectangle is exactly where we expect the projection image appears on the screen finally. Unlike [2] which uses tilt sensors to align the rectangle in horizontal, our method does not have to do so. The projection region of interest can be adjusted to the most suitable viewing direction by the user, making it the best choice in a mobile scenario.

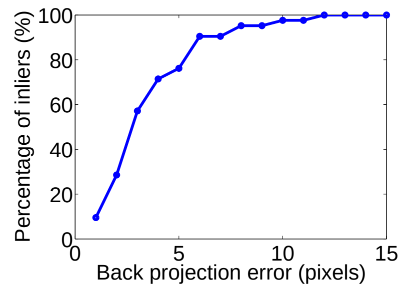


Fig. 3. The distribution of back-projection errors.

C. Pre-warping projection image

By substituting the four corners of the rectangle computed in the previous step into Eq. (1), we can obtain their corresponding points in the projection image. The region enclosed by the four points then becomes the effective projection region, and correcting the keystone effect is done by warping the original display image into this region. To perform the pre-warping, we use a similar homography mapping as in [2] to map the original display image into this effective region. The homography can be calculated from the correspondences between four corners of the effective region and the original display image.

VI. Experimental results

A prototype system is built according to our proposed method. The testing platform is a computer installed with a 2.16GHz dual core processor and 1GB memory stick. The projector-camera pair is comprised of a Optoma mobile projector with resolution of 1280×1024 and a Logitech Quickcam Pro 4000 webcam with resolution of 320×240.

A. Projector-camera calibration

A cardboard with size of 200×150 mm is used to collect correspondences. By changing the position and orientation of the cardboard, totally 42 correspondences are collected to estimate the projection matrix. In order to compensate detection error of the cross and obtain a stable solution, we use a RANSAC estimation scheme in our algorithm. For each run of RANSAC, we randomly select 6 correspondences to estimate the projection matrix. The estimate with most inliers is then accepted as the final result. The accuracy of the estimated projection matrix is measured by the distribution of the back projection error, which is the percentage of the points with back-projection error below some pixel level (inliers). The evaluation is conducted on another stand-alone correspondence set. The error distribution is shown in Figure. The back-projection error corresponding 80% inliers is 2.8 pixels. It is an acceptable accuracy for our general projection application.

B. Projection region tracking

To evaluate the performance of the particle filter tracking of the projection region, we synthesize a random motion trajectory of the projector relative to a virtual wall. The projector

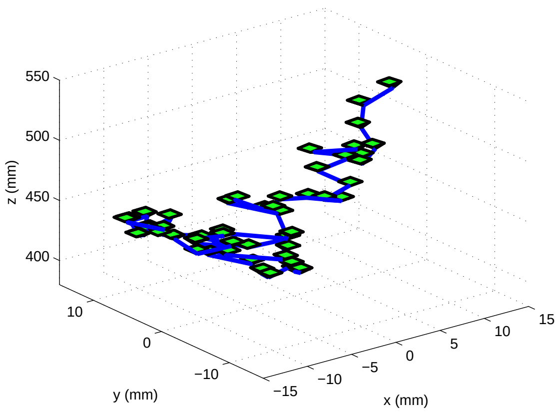


Fig. 4. The recovered trajectory of the projector.

screen is projected onto the wall and then back-projected to the camera, with a standard deviation 2 gaussian noise added. A video sequence of 320 frames containing different poses is created. We run our algorithm on the synthetic video and evaluate the error between the tracked quadrangle and the synthetic ground-truth. Table I lists the mean and std error.

The tracking performance on real data is also tested. A video sequence of 260 frames containing free movements is recorded to evaluate the tracking accuracy. We manually label the position of the quadrangle, and evaluate the error between the tracking results and the manually labeled positions. The mean and std error are listed Table I. We also recovered the trajectory of the projector, as shown in Figure 4. From the experiment, we can see that the algorithm can track the projection region with good accuracy and robustness in both synthetic and real scenarios.

TABLE I
ACCURACY OF THE TRACKING WITH TWO CONFIGURATIONS

	mean (pixels)	standard deviation (pixels)
synthetic	2.8	2.9
real	3.4	4.0

C. Keystone correction

In our experiment, the user casually poses the projector-camera pair and project an image on an ordinary flat surface. It can be clearly observed that the projection content of interest resides in a rectangular area, as shown in Fig. 5. When the user moves the projector around and freely adjusts its pose, our system can still effectively correct the keystone distortion. More results can be found in the supplementary video (or also can be watched online at :).

D. Speed

Our system can achieve a frame rate about 16 fps in our platform. The per-frame processing time is about 60 ms. The pre-warping step occupies most of the time (about 45 ms) due to the large resolution of projection image. With a small projection resolution, the processing time will be dramatically reduced, making our method fast enough for a real time application.

Fig. 5. Some projection results during the moving projection process.

VII. CONCLUSION

In this paper, a camera-projector pair is employed to build a mobile projection system with continuous real-time keystone correction. Since our calibration method and correction mechanism are screen independent, no special display screen is needed for our system and the user can freely project the content onto where he or she likes. Mobility is the most distinguishing feature of our system, while experiment result has also proved its accuracy and real-time processing capacity. As a result, our prototype system is especially suitable for products like integrated camera-projector pair or mobile phones with camera and projector on it.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] R. Sukthankar, R. Stockton, and M. Mullin, “Smarter presentations: Exploiting homography in camera-projector systems,” in *Intl. Conf. on Computer Vision*. IEEE, 2001.

[2] R. Raskar and P. Beardsley, “A self-correcting projector,” in *Intl. Conf. on Computer Vision and Pattern Recognition*, 2001, pp. 504–508.

[3] B. Li and I. Sezan, “Automatic keystone correction for smart projectors with embedded camera,” in *Intl. Conf. on Image Processing*, 2004, pp. 2829–2832 Vol. 4.

[4] M. Leung, K. Lee, K. Wong, and M. Chang, “A projector-based movable hand-held display system,” in *Intl. Conf. On Computer Vision and Pattern Recognition*, 2009, pp. 1109–1114.

[5] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 22, no. 11, 2000.

[6] levmar, “levmar:levenberg-marquardt nonlinear least squares algorithms in c/c++,” *http://www.ics.forth.gr/ lourakis/levmar/*.