

A Tangible Game Interface Using Projector-Camera Systems

Peng Song, Stefan Winkler, and Jefry Tedjokusumo

Interactive Multimedia Lab
Department of Electrical and Computer Engineering
National University of Singapore (NUS)
Singapore 117576
{psong,winkler,jefry}@nus.edu.sg

Abstract. We designed and implemented a tangible game interface using projector-camera systems. The system offers a simple and quick setup and economic design. The projection onto a paper board held by the user provides more direct viewing as well as more natural and flexible interaction than bulky HMD's or monitor-based game interfaces. Homography calibration techniques are used to provide geometrically compensated projections on the board with robustness and accuracy.

Key words: Projector-camera systems, projector-based display, interface design, tilt-board, homography calibration, augmented reality, tangible user interface (TUI).

1 Introduction

With the prevalence of low-cost projectors and cameras, more and more projector-based entertainment systems come into existence. Traditional displays such as CRT monitors and LCD panels are limited by their display size and heavy weight. For large-scale display walls, projectors provide better display quality compared to screens because of the lack of physical seams. However, entertainment activities involving projectors are usually watching movies, playing video games, etc. which only exploit the projector's characteristics in creating large displays. Cameras, if combined with projectors, can enable an intelligent projection system to project certain content and "see" it at the same time. This empowers a projector-camera system to support more complex interactions with the virtual and real world.

A projector's capability of projecting images onto objects or surfaces, combined with a camera's capability of seeing the objects and surfaces, enables the use of projector-camera systems in a variety of augmented reality applications. The early Everywhere Displays [1, 2] and iLamps [3] augment the real world with a projector-camera system, but there is very little user interaction. These two systems are mainly designed to help label the real world and simulate the texture/effects on real world objects respectively. The more recent active pursuit tracking projector-camera system [4] is designed to help humans interact with

volumetric MRI data, while the handheld projector [5] is aimed at both labeling objects and playing games. However, holding a projector in the hand still hinders the user’s interaction in spite of the relative lightness of the portable system. Motion-intensive games cannot be played on handheld projectors, which limits the user’s gaming experience.

An alternative interaction approach is mixed reality games where users have to wear head-mounted displays (HMD’s) or view monitors [6] while moving objects to play the game. HMD’s provide 3D views to the user but affect the comfort and flexibility of players. Besides, they are quite expensive and hard to set up in a new environment. Object-handling interactive games such as Magic Cube [7] provide more flexibility to the user, and the cost is also quite low. However, the separation of the display on the monitor from the object held by the user requires good hand-eye coordination and is not intuitive. Other tangible user interface (TUI) applications such as Geniebottles [8] require active sensors, which are more intrusive in terms of setup and use.

We have designed and implemented a new tangible and interactive interface for games using a projector, a camera and a paperboard. Our system has several advantages:

- *Economic design.* Our system comprises a paperboard together with a cheap off-the-shelf webcam and a projector of any type, all of which are readily accessible; no extra devices, such as sensors, HMD’s, or cubes are needed.
- *Easy setup.* The setup and calibration are simple and quick. A normal home-user pc is sufficient to run the application, as it is computationally inexpensive.
- *Direct viewing.* The games are directly projected onto a tilt board, which provides real-time undistorted views, unlike traditional mixed reality games, where the display is separate from the object.
- *Natural and flexible interaction.* The interactions include natural manipulations of the tilt board to complete different tasks. Holding and manipulating a lightweight cardboard gives the user comfort and flexibility in playing games.

In Section 2, the design of our system is elaborated, followed by a discussion of the system calibration in Section 3. The implementation and results are presented in Section 4.

2 Interface Design

Projector and camera are mounted on tripods. The cardboard is held in the field of view of the camera and the projector. Since the board is light and can be manipulated easily by the user, this tangible interface provides direct viewing and interactive capability. Users move the board for interaction and watch as the projected content changes accordingly.

As shown in Figure 1, the camera is overlooking the projected content on the cardboard. The geometric distortion is captured by the camera and compensated in real-time such that the game interface shown on the cardboard is

always undistorted. The geometric compensation is achieved by homography transformation. Fiducial markers are placed on the paper board for tracking its 3D orientation.

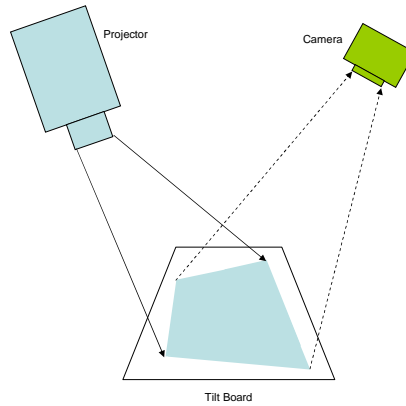


Fig. 1. System design. The projector is mounted above and projecting onto the tilt board. The projection area is observed by the camera. The tilt board acts as the user interface as well as the display.

Our system only requires a cheap commodity projector with an off-the-shelf webcam. Users need not wear or hold any special devices other than a piece of paper to play games on, hence it provides direct, natural and flexible interactions between the player and the computer.

A single projector-camera pair can be used for single-user games, while multiple projectors and cameras can be used to support multi-player games either in a collaborative or competitive fashion. An example of a multi-player projector-camera setup is shown in Figure 2.

3 Geometric Calibration

While playing the game, users have to tilt the cardboard for interaction, which results in oblique projection from the projector. In order to compute the orientation of tilt board and to compensate for the geometrically distorted content projected on the tilted board, geometric calibration techniques are used. Homography calibration is used to compensate for the geometric distortion caused by oblique projection on to the board, and marker tracking is used for estimating the pose of the board.

3.1 Homography Calibration

In an obliquely positioned projector-camera system setup, the geometric relationship needs to be recovered to correctly display an image for the viewer,

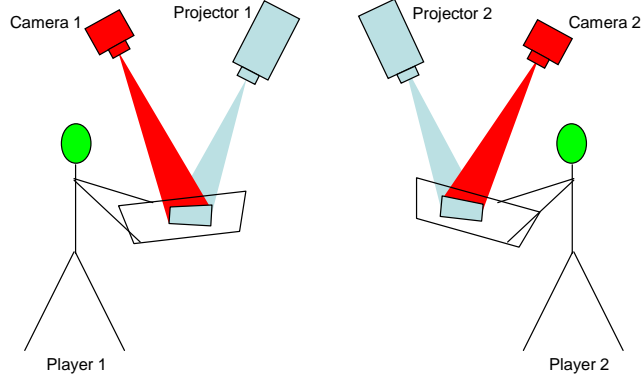


Fig. 2. Multi-player setup. By replicating multiple projector-camera pairs, multi-player games can be played on our interface either in a collaborative or competitive fashion.

otherwise the image shown on the board will be severely distorted, as shown in Figure 3.

As the distortion is predominantly the result of perspective projection, pre-warping the image via a homography can be used to correct for the distortion. The homographies between projector, camera, and the board need to be calibrated prior to pre-warping. Sukthankar *et al.* [9, 10] proposed an automatic keystone correction method for the above setup. Assuming that the pinhole model may be used for projectors and cameras, the transforms between the projector, the camera, and the tilt board can all be modeled as 2-D planar homographies:

$$\begin{bmatrix} kx \\ ky \\ k \end{bmatrix} = \begin{bmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ t_7 & t_8 & t_9 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (1)$$

where (x, y) and (X, Y) are corresponding points in either two frames of reference, and $\mathbf{t} = (t_1, \dots, t_9)^T$ with $|\mathbf{t}| = 1$ are the parameters of the homography. Thus the homography here only has 8 degrees of freedom and can be recovered from as few as 4 pixel correspondences. If more than 4 pixel correspondences are available, more accurate solutions can be determined by a least-squares method. Given m point correspondences, we can have the following $2m \times 9$ matrix \mathbf{A} :

$$\begin{bmatrix} X_1 & Y_1 & 1 & 0 & 0 & 0 & -X_1x_1 & -Y_1x_1 & -x_1 \\ 0 & 0 & 0 & X_1 & Y_1 & 1 & -X_1y_1 & -Y_1y_1 & -y_1 \\ X_2 & Y_2 & 1 & 0 & 0 & 0 & -X_2x_1 & -Y_2x_2 & -x_2 \\ 0 & 0 & 0 & X_2 & Y_2 & 1 & -X_2y_2 & -Y_2y_2 & -y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_m & Y_m & 1 & 0 & 0 & 0 & -X_mx_m & -Y_mx_m & -x_m \\ 0 & 0 & 0 & X_m & Y_m & 1 & -X_my_m & -Y_my_m & -y_m \end{bmatrix}$$

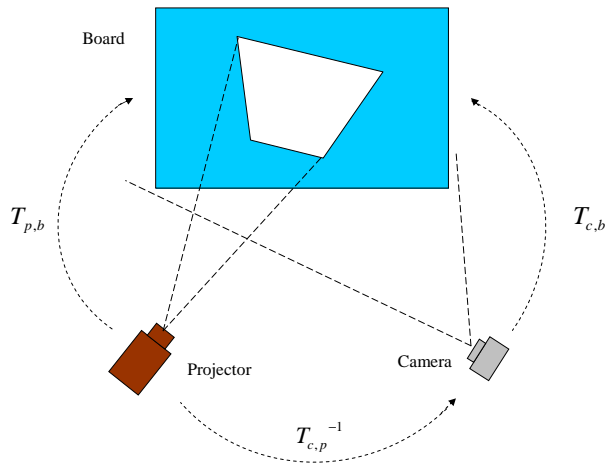


Fig. 3. Homography calibration. Due to misalignment of projector and projection surface, the rectangular display appears as a distorted quadrilateral in the center of the board. This can be compensated by pre-warping using the homographies shown.

Then the unit vector \mathbf{t} needs to be found that minimizes $|\mathbf{A}\mathbf{t}|$. This can be solved by finding the eigenvector corresponding to the smallest eigenvalue of $\mathbf{A}^T\mathbf{A}$.

A simple solution using four feature points correspondences is shown in Figure 3. If a white rectangle is projected onto the board against a high-contrast background, the four corners of the projected quadrilateral in the camera image can be computed. To do this, we extract the contours in the image and check for closed contours with four corners. With the coordinates of the corners known in the projector reference frame, the projector-camera homography \mathbf{T}_{cp} can be recovered. The corners of this quadrilateral completely specify the camera-board homography \mathbf{T}_{cb} . The homography between the projector and the board \mathbf{T}_{pb} can be recovered from the following relationship:

$$\mathbf{T}_{pb} = \mathbf{T}_{cp}^{-1}\mathbf{T}_{cb} \quad (2)$$

By applying the inverse transform \mathbf{T}_{pb}^{-1} to pre-warp the original image, a corrected image can be displayed on the tilt board.

3.2 Tilt Board Pose Estimation

Markers are printed on the board to help track its 3D pose [11]. The image of a marker observed by the camera is a projective transformation of the original physical marker, which includes rotation, translation and scaling. An example is shown in Figure 4.

After thresholding the image observed by the camera, lines and corners are extracted using contour detection. The distorted subimage within the square region can be matched with the original marker subimage through normalization

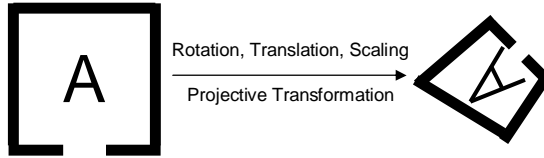


Fig. 4. Projective transformation of a marker. A square marker observed by the camera is a projective transformation of the original marker.

after it has been warped using the homography estimated between the camera and the marker.

The pose of the marker with respect to the camera can be recovered using camera calibration techniques. A point in camera pixel coordinates (x_c, y_c) and in camera coordinates (X_c, Y_c, Z_c) can be related to its board coordinates (X_b, Y_b, Z_b) as:

$$\begin{bmatrix} hx_c \\ hy_c \\ h \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \mathbf{K} \cdot \mathbf{H} \begin{bmatrix} X_b \\ Y_b \\ Z_b \\ 1 \end{bmatrix} \quad (3)$$

where h is an unknown scaling factor. \mathbf{K} and \mathbf{H} are the camera intrinsic and extrinsic matrices, respectively:

$$\mathbf{K} = \begin{bmatrix} s_x f & m & x_0 & 0 \\ 0 & s_y f & y_0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where f is the focal length, s_x, s_y are the scale factors of pixel/mm in x and y directions respectively, m is the slant between x -axis and y -axis (normally zero), (x_0, y_0) is the principal point shift of the camera, (R_{11}, \dots, R_{33}) is the rotation matrix, and $(T_x, T_y, T_z)^T$ is the translation vector.

Pairs of (x_c, y_c) and (X_b, Y_b, Z_b) allow us to estimate the matrices \mathbf{K} and \mathbf{H} using camera calibration techniques. Since standard camera calibration techniques require special physical patterns, we use the MXR Toolkit [12], which was designed specifically for marker tracking.

4 Implementation and Results

Our system comprises an Optoma EP-739H DLP portable projector, a Logitech webcam and a piece of cardboard with square markers printed on it, which is shown in Figure 5. Geometric calibration results and an exemplary racing-ball game running on our interface are described in the following.

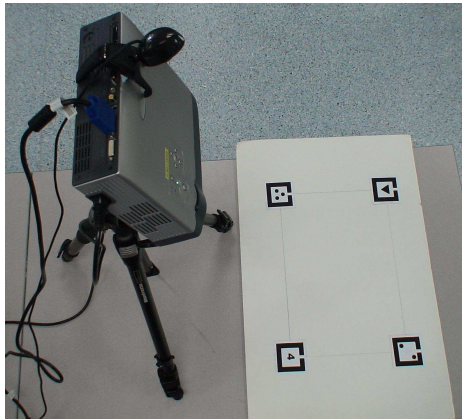


Fig. 5. Tilt board interface using a projector-camera system.

4.1 Geometric Calibration Results

In the geometric calibration of our projector-camera system, the projector-board homography \mathbf{T}_{pb} and camera-board homography \mathbf{T}_{cb} will change as the tilt board moves. The projector-board homography can be estimated through observing projected patterns from the camera and extracting the point correspondences between the camera and the projectors. The camera-board homography \mathbf{T}_{cb} is recovered by observing the physical markers attached on the tilt board with the camera. Subsequently, the projector-board homography can be computed through Eq. (2).

As the player is moving the board constantly, these homographies have to be computed in real-time. However, since the projector and the camera are stationary, we can pre-compute projector-camera homography \mathbf{T}_{pc} from several poses of the board in the system initialization once and for all. This is better than computing \mathbf{T}_{pc} real-time in the game, as this may not be accurate due to camera noise. Furthermore, when the projected area intersects with the physical markers, corner extraction may fail, resulting in a wrongly computed projector-board homography \mathbf{T}_{pb} , which can be avoided by using the pre-computed homography.

Table 1 shows an example of pre-calibrated and real-time computed \mathbf{T}_{pc} , together with the matrix norm differences between the pre-calibrated $\mathbf{T}_{pc,i}$ ($i = 0$) and each real-time computed $\mathbf{T}_{pc,i}$ ($i = 1, 2, 3, 4$). The matrix norm differences between 41 and 172. These differences are contributed mostly by the matrix elements in column 3, which are responsible for the translation, so even though these numbers are large, it has little impact on the shape of the image.

The four corners on the tilt board are also computed using $\mathbf{T}_{pc,0}$ and $\mathbf{T}_{pc,i}$. Then we calculate the corner angles of the quadrilateral at each corner. Figure 6 shows the angle differences of the four corners calculated for four arbitrary tilt board poses. Take note that even though pose 1 has the smallest norm difference, it has the largest difference in degree. This is mainly due to the difference between

elements in column 1,2 of $\mathbf{T}_{pc,1}$ and $\mathbf{T}_{pc,0}$, which are larger compared to other poses. To conclude, pre-calibrated \mathbf{T}_{pc} may not be 100% accurate, but it is more robust and stable when projected corners overlap with physical markers.

Table 1. Pre-calibrated and real-time computed \mathbf{T}_{pc}

| | $\mathbf{T}_{pc,i}$ | Matrix Norm (real-time vs. pre-computed) |
|------------------|--|---|
| Pre-computed | 1.452257 0.061707 -500.917816 -0.007086 1.234739 -262.956879 -0.000012 -0.000341 1 | 0 |
| Real-time Pose 1 | 1.390219 -0.05998 -462.887207 -0.041811 1.457855 -279.226654 0.000046 -0.000807 1 | 41.3651 |
| Real-time Pose 2 | 1.545433 0.081606 -550.048706 0.002831 1.286384 -246.801178 -0.000019 -0.000303 1 | 51.719 |
| Real-time Pose 3 | 1.606216 0.108573 -571.732544 -0.002608 1.242908 -244.1651 0.000018 -0.000133 1 | 73.2658 |
| Real-time Pose 4 | 1.854606 0.096473 -660.308655 -0.223254 1.434345 -196.871918 0.00035 -0.000304 1 | 172.5482 |

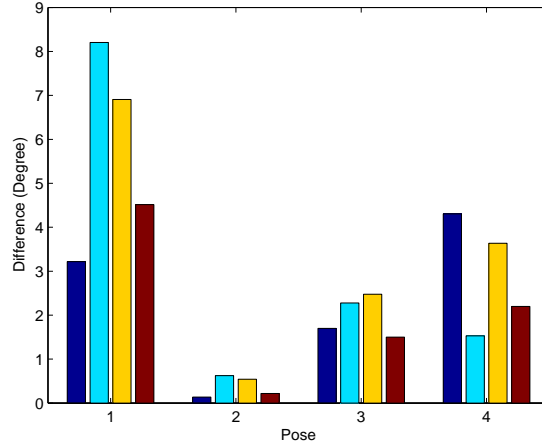


Fig. 6. Differences (in degree) of projected corners using precalibrated vs. real-time computed \mathbf{T}_{pc} for 4 different board poses.

4.2 Racing-Ball Game

We implemented a racing-ball game on our system, as shown in Figure 7. The player tilts and slants the board to steer the ball through a maze while avoiding holes and collecting bonus items.

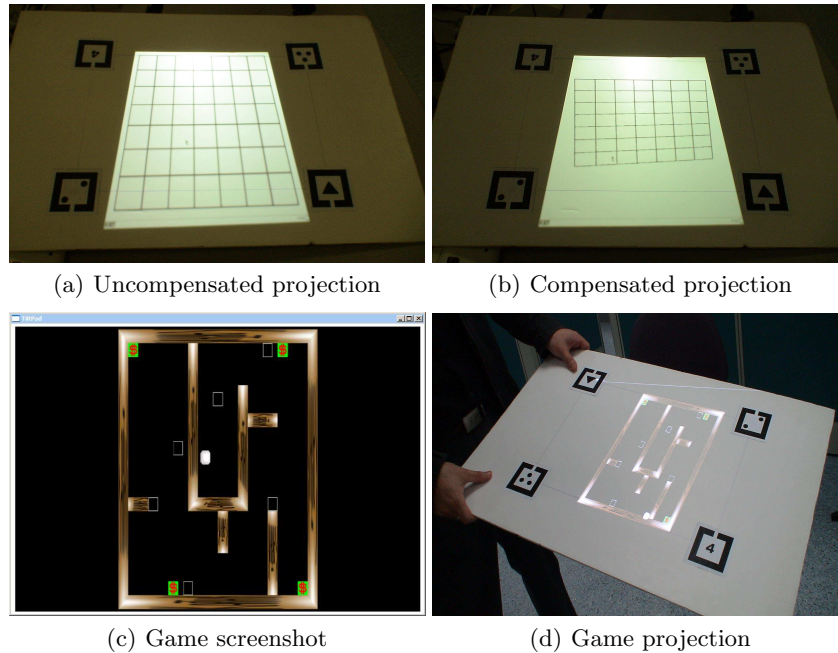


Fig. 7. Racing-ball game implementation using cardboard and projector-camera system. (a) geometrically uncompensated projection; (b) geometrically compensated projection; (c) original racing-ball game screenshot; (d) in-game shot of user slanting and tilting the cardboard in the racing-ball game.

5 Conclusions and Future Work

We have designed and implemented a tangible game interface using a projector-camera system. Our interface provides a more direct and natural interaction than HMD's or monitor-based interfaces. Homography calibration and physical marker tracking are used to implement the system. The system setup is simple, easy, quick and economic since no extra special devices are needed.

We are now conducting a usability study to evaluate the system from a user perspective. Furthermore, due to the oblique projection, there are issues such as projection color imbalance and out-of-focus projector blur that need to be investigated.

References

1. Pinhanez, C.: The Everywhere Displays projector: A device to create ubiquitous graphical interfaces. In: Proc. Ubiquitous Computing, Atlanta, GA (2001) 315–331
2. Pingali, G., Pinhanez, C., Levas, A., Kjeldsen, R., Podlaseck, M., Chen, H., Sukaviriya, N.: Steerable interfaces for pervasive computing spaces. In: Proc. IEEE International Conference on Pervasive Computing and Communications, Fort Worth, TX. (2003) 315
3. Raskar, R., Baar, J., Beardsley, P., Willwacher, T., Rao, S., Forlines, C.: iLamps: Geometrically aware and self-configuring projectors. *ACM Transactions on Graphics* **22**(3) (2003) 809–818
4. Gupta, S., Jaynes, C.: Active pursuit tracking in a projector-camera system with application to augmented reality. In: IEEE International Workshop on Projector-Camera Systems (PROCAMS). (2005) 111–119
5. Beardsley, P., van Barr, J., Raskar, R., Forlines, C.: Interaction using a handheld projector. *IEEE Computer Graphics and Applications* **25**(1) (2005) 39–43
6. Cheok, A., Fong, S., Goh, K., Yang, X., Liu, W., Farbiz, F.: Human Pacman: A sensing-based mobile entertainment system with ubiquitous computing and tangible interaction. In: 2nd Workshop on Network and System Support for Games. (2003) 106–117
7. Zhou, Z., Cheok, A., Pan, J., Li, Y.: Magic Story Cube: An interactive tangible interface for storytelling. In: Proc. ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, Singapore (2004) 364–365
8. Mazalek, A., Wood, A., Ishii, H.: Geniebottles: An interactive narrative in bottles. In: SIGGRAPH Conference Abstracts and Applications, Los Angeles, CA (2001) 189
9. Sukthankar, R., Stockton, R., Mullin, M.: Automatic keystone correction for camera-assisted presentation interfaces. In: Proc. International Conference on Multimodal Interfaces, Beijing, China (2000) 607–614
10. Sukthankar, R., Stockton, R., Mullin, M.: Smarter Presentations: Exploiting homography in camera-projector systems. In: Proc. International Conference on Computer Vision, Vancouver, Canada (2001) 247–253
11. Kato, H., Billinghurst, M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: Proc. 2nd International Workshop on Augmented Reality (IWAR), San Francisco, CA (1999) 85–95
12. NUS Mixed Reality Lab: MXR Toolkit. <http://mxrtoolkit.sourceforge.net/>