Zhaorong Li , KH Wong, MC Leung, HF Ko,
KK Lee and MMYuen Chang, Multimedia Systems,
Springer, Volume 17, Number 5,
435-447, February 2011.

A Projector-based Movable Hand-held Display System for Interactive 3D Model Exhibition

Authors

***Abstract*—Traditional display systems usually display objects on static screens (monitors, walls etc) and the interaction between the displaying object and the viewer can only be via keyboard and mouse. It will be attractive if we display the object to a hand-held screen and interact with it using our hands as we do in our daily lives. In this paper, we propose a prototype system by projecting the object to a hand-held foam sphere. The target is to develop an interactive 3D model exhibition tool without the viewer having to wear spectacles. In our system, the viewer holds the sphere with his hands and moves it freely. Meanwhile we project well-tailored images onto the sphere coincident with the movement of it, giving the viewer a virtual perception as if the object is sitting inside the sphere and being moved by the viewer. The design goals of our system are *low-cost, real-time, live*, and *3D*. An off-the-shelf projector-camera pair is first calibrated via a simple but efficient algorithm. Vision based algorithms are proposed to detect the sphere and track its subsequent motion. To adapt to different application scenarios, we develop two kinds of configurations to track the sphere. The projection image is generated based on the projective geometry among the projector, sphere, camera and the viewer. We discuss how to allocate the view spot and warp the projection image. We also present the result and the performance evaluation of the system.**

***Index Terms*—Virtual and augmented reality, 3D interactive display, projector-camera system, object tracking.**

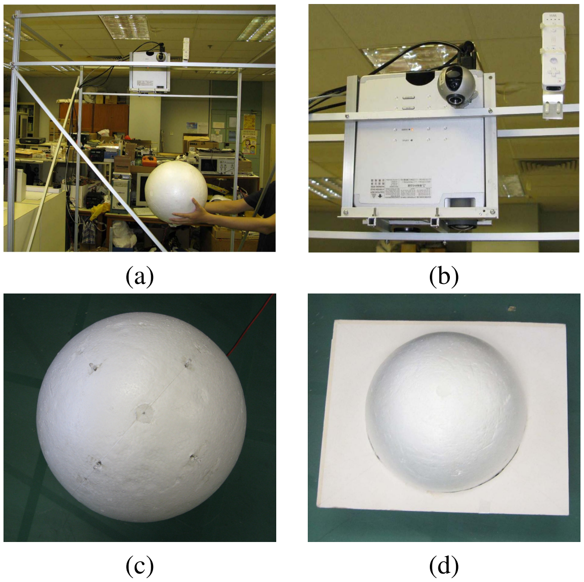


Fig. 1. The configuration of our system: (a) The rig; (b) The projector, camera and Wiimote; (c) The sphere with 4 IR LEDs. (d) The sphere and the cardboard.

I. Introduction

TRADITIONAL 2D display systems usually display objects on static screens and the viewer interacts with it using keyboard and mouse, thus having the disadvantage of unnatural user interaction and low perceived level of reality. Projector is a good choice to improve the freedom and interaction of these systems, but existing ones usually display 2D information.Moveover, special hardware such as magnetic sensors are usually included in these system. Polarization techniques are popular to create 3D perception, but it require the viewer to wear specially-designed spectacles. It is still challenging to develop an interactive 3D display system in low-cost and easily-built fashion. As an attempt, this paper proposes a system for interactive 3D object exhibition, in which we display the 3D model of the object to a hand-held screen and we can interact with it using own hands directly as we do in our daily lives.

The proposed system finds many applications in real life. For instance, in a museum mounted with many projectors, the visitor can use this hand-held sphere to explore the computer-generated copy of the original relics which are not available to the visitor for touching. In this way, the visitor have a realistic feeling about the relics just like holding it in the hands while the relics is protected from damage. Manufacturers can also use this tool to examine a new product which is still in the design stage and a real product is not available. It will help the designer to have a better evaluation of the product before it is put on stream.

The objective of our work is to build a *low cost, easily-built* and *workable* 3D interactive object exhibition tool without the viewer to wear spectacles. Instead of using magnetic sensors or specially-designed hardwares, we use several low-cost off-the-shelf devices and computer vision techniques to build the system. The main idea of the system is to project the displaying object onto a hand-held sphere. When the viewer moves and rotates the sphere, we use object tracking techniques to track the translation and rotation of the sphere. Meanwhile, based on the pre-calibrated projective geometry among the camera, sphere, projector and the viewer, we project well-tailored images of the object onto the sphere depending on the translation and rotation of it. By continuously adjusting the images projected to the sphere as it moves and rotates, the motion parallax gives the viewer a virtual 3D perception as if the object is sitting inside the sphere and being moved and rotated by the user directly. This gives the viewer a direct, natural and realistic experience. The devices used in our system include a projector, a webcam, a Nintendo Wiimote and a foam sphere. To adapt to different application scenarios, we design two kinds of configurations to track the sphere. In the first configuration, we embed four IR LEDs on the

Manuscript received October, 2009.

surface of the sphere, and include a Nintendo Wiimote to track their positions. In the second configuration, the sphere is encompassed in the center of a cardboard. In Fig. 1 (a), we show the whole picture of our system. The camera, projector, and Wiimote are fixed on a rig, shown in (b). In Fig. 1 (c),(d), we show two configurations of the sphere.

The development of the system faces challenges in various computer vision and graphic fields, including projective geometry, projector-camera calibration, object tracking, and spherical display. Our contribution is mainly the proposal of a new type of display system, and the system integration of different technologies and devices. The remainder of the paper presents how we handle different challenges. It is organized as follows. Section II discusses some related work. Section III gives the overview of the system. Section IV describe the calibration of the projector-camera pair. In section V, we introduce how to track the translation and rotation of the sphere. Section VI describe how to generate the projection image correctly. The implementation details and results are given in Section VII. We conclude the paper in Section VIII.

II. Related Work

Projector based system is not new. It is popular in Augmented Reality (AR) and Human Computer Interaction (HCI) since such systems improve the freedom of the display and provide easy ways of man machine interface. According to the mobility of the projectors and screens, projector based systems mainly fall into two categories: static or movable. In this section, a review to existing projector based systems is presented. Especially, two types of systems closely related to the proposed one, curved-surface systems and movable-surface systems, are discussed in more detail.

A. Projector Camera System

In most of the traditional applications of projector-based systems, the projectors and the screens are at fixed positions. One popular application is to use multiple projectors to build large display walls for creating immersive environment. The CAVE system[1] uses three rear projectors to project onto three walls of a cube-shaped room and one down projector to project onto the floor, creating a fully immersive virtual reality environment. The Teleport system[2] uses a projected wall to create the illusion of extending the room to another one for a teleconferencing system. Bimber *et. al*[3] proposes a view-dependent stereoscopic projection system which can project display content onto a natural wall. Through geometric and photometric correction, images projected onto a rough wall look similar to those on a flat screen in the view of the user. Projection technology are also used to modify the appearance of a real object or an environment. The Shader Lamp[4] explores the use of projection light to alter the appearance of a complex 3D object. The ability of controlling the appearance of an object enables applications such as simulating a real scene[5], making one object looks like another[6], or enhancing the appearance of the original object[7].

In order to create the correct projection, projector-based systems require various calibration technologies , including geometric calibration, photometric calibration etc. Since the projector cannot observe the projection result, cameras are usually used as a visual feedback for the calibration. When the projection surfaces deviate from a plane or the projection to the screen is oblique, the projected images will be geometrically distorted. For a planar screen, the distortion is known as the keystone distortion. The projector camera mapping can be represented by a 3x3 homograph matrix in the planar case. Sukthankar *et al.*[8] proposed a smart presentation system, in which the homography matrix is calibrated via the point correspondences between the projection and camera image plane, and accordingly the keystone distortion is corrected. Self-correcting projector system[9] and autocalibration algorithms[10] without using markers, as well as multiple projectors calibration[11] are also proposed. Apart from geometric calibration, photometric stiching is necessary in a multiple projector system. Discussion can be found in [12][13].

B. Curved Display Surface

Curved display surfaces are mainly used in two kinds of applications. One is for large scale curved display which gives the viewer an immersive experience and more freedom of view compared with planar surfaces. Multiple projectors and cameras are usually included in such systems in order to cover a larger surface. For example, Raskar *et al.* [14], [15] proposed a scalable panoramic display system with multiple casually positioned projectors. The geometric calibration of the projector-camera pair with curved surface is more complicated since their correspondence is no longer a homography. In [15], Raskar proposed a parametric approach called the quadric transfer to represent the correspondence for quadric surface. We do not employ this method in our system but adopt another approach (detailed in Section III) which is simple but efficient in locating the points on the movable sphere surface. Another kind of application is for non-regular surface display. Kondo *et al.* [16] proposed a Free Form Projection Display (FFPD) system for displaying images on arbitrarily-shaped surfaces. By scanning the 3D structure of the surface with a 3D scanner, they can display 3D content to the surface without distortion. One main application of this system is for medical education as demonstrated in an extended work[17]. Lee *et al.* [18] proposed an algorithm to display on some regularly-foldable surfaces, such as scroll, fan, and umbrella.

C. Moveable Surface

Mobility plays a more and more important role in the development of projector-based applications. With a dynamic projector or screen, more interaction can be introduced in the system which greatly enriches the user experience. The success of movable display surface system relies on a reliable tracking algorithm of the surface. Two major categories tracking systems are used, sensor based and camera based. The Dynamic Shader Lamp[19] extends the previous work in [4] to allow users to hold the objects in their hands by adding a six degree of freedom optical tracker and a magnetic tracker. The object used as the display surface in [16] is also movable, whose movement is tracked using magnetic sensors. Lee *et*

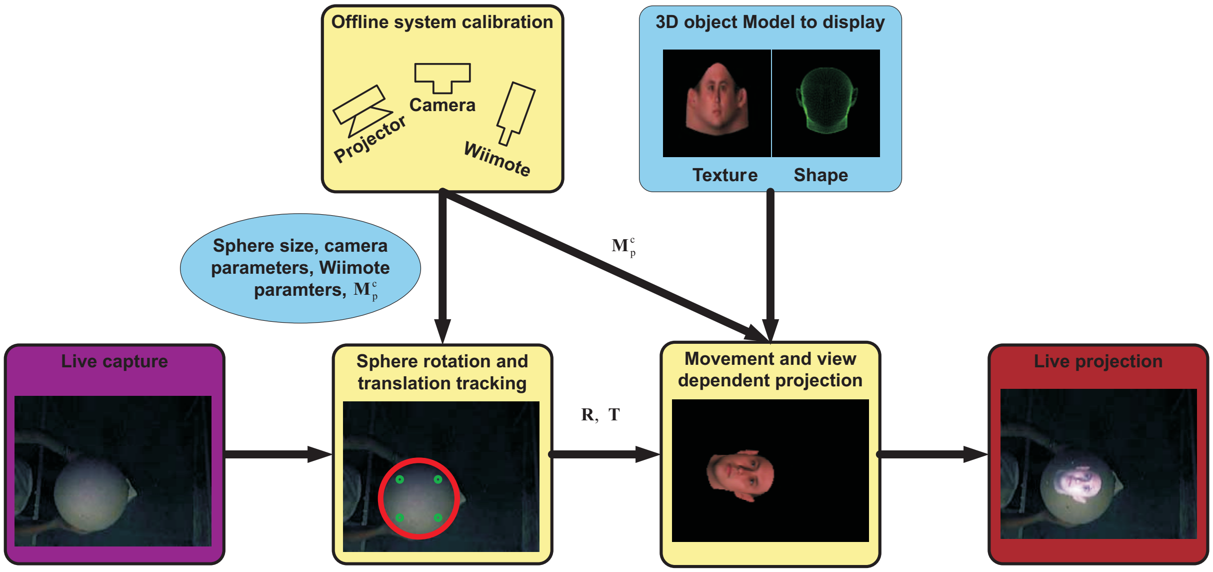

Offline system calibration
Camera
Projector
Wiimote
3D object Model to display
Texture
Shape
Sphere size, camera parameters, Wiimote paramters, $\mathbf{M}_{\mathrm{p}}^{\mathrm{c}}$
$\mathbf{M}_{\mathrm{p}}^{\mathrm{c}}$
Live capture
Sphere rotation and translation tracking
R, T
Movement and view dependent projection
Live projection

Fig. 2. The overview of our system.

al[20] proposed to use the projector itself as a tracker of the surface to eliminate an external tracking system, however it is limited to DLP projectors. On the other hand, camera is usually employed to track simple and regular surfaces using computer vision techniques. As a camera is usually included to calibrate the projector, it is natural to use it to track the display surface without including extra ones. The main advantage of camera tracker over sensor tracker is its low cost, though maybe at the expense of a decrease in accuracy and robustness. The Portable Display Screen (PDS) system[21] detect and track a cardboard with black borders using Hough transform and Kalman filter. Gupta *et al* [22] proposed an Active Pursuit Tracking algorithm in which four color fiducials are attached onto a white cardboard and these fiducials are tracked using Camshift algorithm. Leung *et al* [23] used a particle filter algorithm to track the cardboard with edge features. Since our design goals of our system is low cost and easy-built, it is based on computer vision technologies.

III. System Overview

The system is an integration of three major modules, the calibration module, the tracking module, and the projection module. Fig. 2 illustrates the interaction between different modules and the input and output. The calibration module finds the relationship between the projector and camera. To register the projection image onto the moving sphere, the system needs to know the position and orientation of the sphere at each time instant. The tracking module tracks the translation and the rotation of the sphere relative to the camera. Based on the pre-calibrated projector-camera relationship and the tracked movement of the hand-held sphere, the projection module generate the projection image of the displaying object and project it onto the sphere. The following sections describe each module in detail.

IV. Projector-Camera Pair Calibration

The target of calibrating the projector camera pair is to use the camera to guide the projection. Previous calibration methods applied to planar surfaces and static systems are no longer applicable due to the movable sphere surface we used. Alternatively, we estimate the projection matrix of a 3D point in the camera coordinate system to the 2D projector image plane. The projection matrix is constant and independent from the movement of the sphere since the projector and camera are fixed. Moreover, it is unnecessary to explicitly estimate the intrinsic parameters of the projector and the relative pose between the projector and camera. This makes our calibration fairly easy. A simple calibration algorithm is proposed. The basic idea is to use the sphere as the calibration object. By manually marking a number of correspondences of the sphere's surface points in image pairs of the projector and the camera, we can estimate the projection matrix.

A. The projector model

The ideal projection model of the projector is the same as that of the camera except the projection direction. The projection from a 3D world point to the 2D projector image pixel is also via a 3x4 perspective projection matrix. We assume the world coordinate identical to the camera coordinate here. Then any 3D point in the camera coordinate, for example, the sphere surface point $\mathbf{P}^{\mathrm{c}}(x, y, z)$, and its projector image pixel, $\mathbf{p}^{\mathrm{p}}(u^{\mathrm{p}}, v^{\mathrm{p}})$ are related by a projection matrix $\mathbf{M}_{\mathrm{p}}^{\mathrm{c}}$:

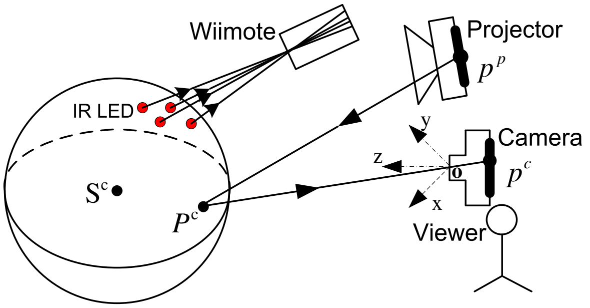


Fig. 3. The projective geometry between the project camera pair.

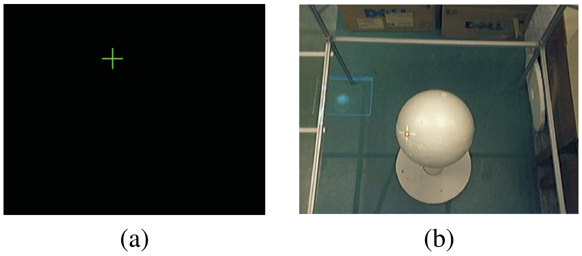


Fig. 4. Finding correspondence between the projector and camera image. (a) The cross image projected to the sphere; (b) The cross image observed by the camera.

$$s\tilde{\mathbf{p}}^{\mathrm{p}} = \mathbf{M}_{\mathrm{p}}^{\mathrm{c}}\mathbf{P}^{\mathrm{c}} \quad (1)$$

and

$$\begin{aligned}\mathbf{M}_{\mathrm{p}}^{\mathrm{c}} &= \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \\ &= \begin{pmatrix} f_{\mathrm{u}} & 0 & u_0 \\ 0 & f_{\mathrm{v}} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{T} \end{pmatrix} \quad (2)\end{aligned}$$

Here, f_{u}, f_{v}, u_0, v_0 are the intrinsic parameters of the projector. $\mathbf{R}$ and $\mathbf{T}$ are the rotation matrix and translation vector from the camera to the projector. In our system, we don't need to explicitly estimate the intrinsic and extrinsic parameters, but instead we only need to estimate the projection matrix $\mathbf{M}_{\mathrm{p}}^{\mathrm{c}}$.

B. Estimation of the projection matrix

The projective geometry of the projector camera pair is shown in Fig. 3. The light from some pixel $\mathbf{p}^{\mathrm{p}}(u^{\mathrm{p}}, v^{\mathrm{p}})$ in the projector image intersects the sphere at $\mathbf{P}^{\mathrm{c}}(x, y, z)$ (in camera coordinate), and then create pixel $\mathbf{p}^{\mathrm{c}}(u^{\mathrm{c}}, v^{\mathrm{c}})$ in the camera. These three points $(\mathbf{p}^{\mathrm{p}}, \mathbf{P}^{\mathrm{c}}, \mathbf{p}^{\mathrm{c}})$ form a correspondence. The basic idea of estimating the projection matrix is to collect a number of such correspondences. We collect each correspondence in this way: we project a cross onto the sphere surface and observe the cross using the camera. An example of the projection cross image and the corresponding image observed by the camera is shown in Fig. 4. The 2D coordinates of the points $\mathbf{p}^{\mathrm{p}}$, and $\mathbf{p}^{\mathrm{c}}$ can be manually labeled while the 3D coordinates of the points on the sphere surface $\mathbf{P}^{\mathrm{c}}$ cannot be directly obtained. In order to calculate $\mathbf{P}^{\mathrm{c}}$, we first need to locate the 3D position of the sphere's center in the camera coordinate.

The 3D position of the sphere can be located based on its image in the camera. According to [24], the image of a sphere is a conic section under the pinhole perspective camera model. Since the depth information is lost in perspective projection, the conic section could be created by a family of center-collinear spheres. Only given the conic section, we cannot uniquely recognize the true sphere out of the family. However, once the physical radius of the sphere is given, we can uniquely locate the sphere. We use the geometric method proposed in [25] to locate the sphere's center. The basic idea of the method is to investigate the relationship between the general case where the sphere locates in arbitrary position and the special case where the sphere lies at some position along the z-axis of the camera. In the special case, the image of the sphere is a circle and the sphere's center can be easily located given the circle. The sphere in arbitrary position can be viewed as rotated from a sphere in the z-axis. Accordingly, the image of the sphere changes from a circle to a conic section due to the rotation. This is illustrated in Fig. 5. So given the conic section, we first regulate it to a circle and obtain the rotation. Then we locate the sphere's center in the special case based on the circle and rotate it to get the sphere's center in the general case with the same rotation. In our implementation, we use the Hough transform circle detection algorithm to detect a circle as the approximation of the conic section.

After the sphere's center is located, we can locate the corresponding point on the sphere surface for each pixel within the conic section. For each correspondence $(\mathbf{p}^{\mathrm{p}}, \mathbf{P}^{\mathrm{c}}, \mathbf{p}^{\mathrm{c}})$, the sphere surface point $\mathbf{P}^{\mathrm{c}}$ in camera coordinate should satisfy the following equations:

$$\begin{aligned} s\tilde{\mathbf{p}}^{\mathrm{c}} &= \mathbf{K}^{\mathrm{c}}\mathbf{P}^{\mathrm{c}} \\ \|\mathbf{P}^{\mathrm{c}} - \mathbf{S}^{\mathrm{c}}\|_2^2 &= R^2 \quad (3)\end{aligned}$$

where s is a scale factor, $\tilde{\mathbf{p}}^{\mathrm{c}}$ is the homogeneous coordinate of $\mathbf{p}^{\mathrm{c}}$, and $\mathbf{K}^{\mathrm{c}}$ is the intrinsic parameter matrix of the camera which is calibrated beforehand using the OpenCV toolbox. The first equation is the projection equation, and the second is to constrain the distance between the surface point and the sphere center. We solve them to obtain $\mathbf{P}^{\mathrm{c}}$ for each correspondence. Note that two solutions can be obtained but we simply discard the one further from the camera center because it is irreasonable.

Now for each calculated correspondence, we can write a projection equation according to Eq. (1) and Eq. (2):

$$s\begin{pmatrix} u^{\mathrm{p}} \\ v^{\mathrm{p}} \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} (4)$$

We rewrite it in the equivalent form by eliminating the scale factor:

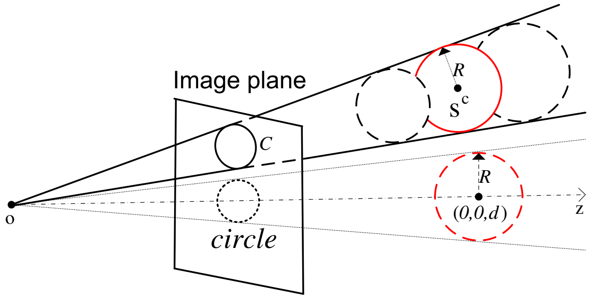


Fig. 5. The center-collinear spheres and their common conic section. The sphere in arbitrary position can be viewed as rotated from a sphere in the z-axis.

$$
\begin{aligned}
u^{\mathrm{p}} &= \frac{m_{11}x + m_{12}y + m_{13}z + m_{14}}{m_{31}x + m_{32}y + m_{33}z + m_{34}} \\
v^{\mathrm{p}} &= \frac{m_{21}x + m_{22}y + m_{23}z + m_{24}}{m_{31}x + m_{32}y + m_{33}z + m_{34}}
\end{aligned} \tag{5}
$$

It can be further re-arranged into the following form:

$$
\begin{aligned}
&xm_{11} + ym_{12} + zm_{13} + m_{14} - u^{\mathrm{p}}xm_{31} \\
&\quad - u^{\mathrm{p}}ym_{32} - u^{\mathrm{p}}zm_{33} - u^{\mathrm{p}}m_{34} = 0 \\
&xm_{21} + ym_{22} + zm_{23} + m_{24} - v^{\mathrm{p}}xm_{31} \\
&\quad - v^{\mathrm{p}}ym_{32} - v^{\mathrm{p}}zm_{33} - v^{\mathrm{p}}m_{34} = 0
\end{aligned} \tag{6}
$$

Assuming that we collect totally n correspondences, $(\mathbf{p}_{\mathrm{i}}^{\mathrm{p}}, \mathbf{P}_{\mathrm{i}}^{\mathrm{c}}, \mathbf{p}_{\mathrm{i}}^{\mathrm{c}})$, $\mathrm{i} = 1 \ldots n$, we rearrange all the equations to a system of the form $\mathbf{Gm} = \mathbf{0}$, where $\mathbf{G}$ is a $2n \times 12$ matrix, $\mathbf{m}$ is a 12×1 vector arrangement of the rows of the projection matrix. There are totally 12 variables, so $n \geq 6$ correspondences are enough to solve it. We obtain a solution which introduces the least error using Singular Value Decomposition (SVD). Moreover, in order to compensate labeling errors and obtain a stable solution, we take following steps: first, we use a RANSAC scheme in our algorithm. For each run of RANSAC, we randomly select 6 correspondences to estimate the projection. The criterion for admitting an inlier is that the sum of the absolute back-projection errors in x and y axis is below 10 pixels. Second, a fine adjustment is carried out on the RANSAC result. It minimize the following sum of the squared back-projection errors:

$$
\begin{aligned}
\sum_{i=1}^{n} (u_i^{\mathrm{p}} &- \frac{m_{11}x_i + m_{12}y_i + m_{13}z_i + m_{14}}{m_{31}x_i + m_{32}y_i + m_{33}z_i + m_{34}})^2 + \\
&(v_i^{\mathrm{p}} - \frac{m_{21}x_i + m_{22}y_i + m_{23}z_i + m_{24}}{m_{31}x_i + m_{32}y_i + m_{33}z_i + m_{34}})^2
\end{aligned} \tag{7}
$$

Taking the RANSAC solution as the initialization, we use the Levenberg-Marquardt method to minimize the error. With these strategies, the accuracy of the estimated projection matrix is further improved.

V. Sphere Detection and Tracking

In our system, we have to detect and track the translation and rotation of the sphere relative to the camera. The translation of the sphere is defined as the position of the sphere's center in the camera coordinate. As for the rotation, since the sphere is centrisymmetric, it is necessary to attach a reference to define it. Here, two configurations to define and track the rotation are proposed. In configuration I, we embed four IR LEDs on the sphere surface and use the PixArt IR camera embedded in the Wiimote to track the LEDs. In configuration II, we encompass the sphere in the center of a rectangle cardboard. The rotation of the sphere is inferred from the orientation of the cardboard. Compared with configuration II, the configuration I has a more user-friendly appearance since the IRLEDs embedded are almost invisible. Moreover, It has a better robustness and accuracy. However, its disadvantage is that it requires an extra IR camera, which will increase the cost and also require a calibration step before using it. The choice of this two configurations depends on the requirement and budget of the applications.

A. Configuration I: Using Embedded LED and Wiimote

In this configuration, four IR LEDs are evenly embedded on the sphere surface. The rotation of the sphere is defined as follows: we define an object coordinate in the center of the sphere. The x-y plane parallels the plane formed by the four IR LEDs. An illustrative figure is shown in Fig. 6 (a). The rotation of the sphere is defined as the rotation from the object coordinate to the camera coordinate.

The IR LEDs and thus the rotation of the sphere are tracked using the Wiimote. However, the rotation tracked is relative to the Wiimote but that we want is relative to the camera, we have to first calibrate the Wiimote camera pair. Similar to calibrating the projector camera pair , we don't explicitly estimate the relative pose between the Wiimote and camera since the explicit calibration of the relative pose is unnecessary, instead we estimate the projection matrix which project a 3D point in the Wiimote coordinate to a 2D pixel in the camera image plane. The idea of the calibration is also to collect a number of correspondences of the IR LEDs in the Wiimote and camera image plane. The IR LEDs' position in the Wiimote image are detected by the Wiimote automatically while we label their positions in the camera image manually. The 3D coordinates of the IR LEDs in the Wiimote coordinate can be calculated via the Perspective Four Points (P4P) algorithm, which estimates the 3D coordinates of four object points with known configurations based on their image pixels in a calibrated camera. The intrinsic parameters of the Wiimote are calibrated using the OpenCV toolbox by regarding the four IRLEDs as a calibration board. The projection matrix $\mathbf{M}_{\mathrm{c}}^{\mathrm{w}}$ is estimated similarly as estimating $\mathbf{M}_{\mathrm{p}}^{\mathrm{c}}$.

1) Detection: We use Hough transform circle detection algorithm to detect a circle to approximate the conic section in the initial frame of the video stream, and employ the algorithm introduced in Section IV to locate the center of the sphere. The rotation is calculated as follows: given the four detected IR LEDs' positions in the Wiimote image, we calculate their 3D coordinates in Wiimote coordinate using the P4P algorithm. The four 3D points are then projected to the camera image plane by the projection matrix $\mathbf{M}_{\mathrm{c}}^{\mathrm{w}}$. Finally we can calculate their 3D coordinates in camera coordinate using Eq. (3) since we have located the center of the sphere. Assuming that they are $\mathbf{L}_{\mathrm{i}}^{\mathrm{c}}, \mathrm{i} = 1 \dots 4$, we can obtain the base vectors of the object coordinate and the rotation matrix from the object to the camera by:

$$\mathbf{i} = \frac{\mathbf{L}_3^{\mathrm{c}} - \mathbf{L}_2^{\mathrm{c}}}{|\mathbf{L}_3^{\mathrm{c}} - \mathbf{L}_2^{\mathrm{c}}|}, \quad \mathbf{j} = \frac{\mathbf{L}_1^{\mathrm{c}} - \mathbf{L}_4^{\mathrm{c}}}{|\mathbf{L}_1^{\mathrm{c}} - \mathbf{L}_4^{\mathrm{c}}|}, \quad \mathbf{k} = \mathbf{i} \otimes \mathbf{j}$$
$$\mathbf{R}_{\mathrm{c}}^{\mathrm{o}} = [\mathbf{i}, \mathbf{j}, \mathbf{k}] \tag{8}$$

We further refine the translation and rotation by minimizing the following squared errors:

$$\sum_{i=1}^{4} \|\mathbf{R}_{\mathrm{c}}^{\mathrm{o}} \mathbf{L}_i^{\mathrm{o}} + \mathbf{T}_{\mathrm{c}}^{\mathrm{o}} - \mathbf{L}_i^{\mathrm{c}}\|_2^2 \tag{9}$$

where $\mathbf{L}_{\mathrm{i}}^{\mathrm{o}}, \mathrm{i} = 1 \dots 4$ are the 3D coordinates of the four IR LEDs in the object coordinate. They are measured beforehand according to the configuration of the LEDs.

2) Tracking: After detecting the the translation and rotation of the sphere in the initial frame, we track it in the subsequent frames. The tracking state is the concatenation of the rotation and translation vector in the following form:

$$s = \begin{pmatrix} r_x & r_y & r_z & t_x & t_y & t_z \end{pmatrix} \tag{10}$$

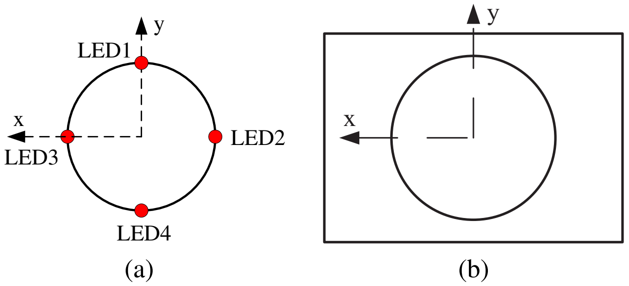
where r_x, r_y and r_z is the Euler angle along the x, y and z axis respectively and t_x, t_y and t_z is the translation along the x, y and z axis respectively.

Particle filter is used to estimate the posterior density of the pose. It represents the pose as a set of discrete particles. Each particle has a weight to indicate how confident it is to represent the pose. The two main components of a particle filter are the state dynamic model and the observation model. The state dynamic model determines how the particles propagate from frame to frame. The observation model determines how much weight is assigned to particles providing the observation at that frame. The workflow of the particle filter used in our system is shown is Fig. 7. We describe the state dynamic model and observation model as follows:

a) State dynamic model: Since the sphere is freely moved, a simple random walk model based on a uniform density U about the previous state is used. The variable e represents the uncertainty about the movement of the sphere.

$$p(s_k|s_{k-1}) = U(s_{k-1} - e, s_{k-1} + e) \tag{11}$$

b) Observation model: The observation in our algorithm is the edge map obtained by Canny edge detector and the detected position of the four IR LEDs. To evaluate the likelihood of each particle, we first re-project the sphere and the four IR LEDs to the image plane according to the pose represented by the particle. The projected sphere is an approximate circle. We check how many edge points are on the circle. A edge point is considered on the circle if its distance to the circle center is within 10 pixel. For each degree of the 360-degree circumference, we check if there is an on-edge point. If the number of the on-edge points is less than 90, i.e, one fourth of the whole circumference, we regard the on-edge points are not enough to match a valid circle, and a very low likelihood is assigned to this particle. Otherwise, we fit several circles centered close to the projected circle based on these on-edge points. The one with the fewest fitting outliers is considered to the best one and its fitting rate (the ratio of the inliers and the total on-edge points) is assigned to the particle as its likelihood. To give a more precise tracking result, we introduced a replacement scheme into our observation model. For particle whose fitting rate is above a threshold (0.6 in our implementation), we relocate the sphere's center based on the best fitted circle and recalculate its rotation based on the detected IR LEDs, and replace the particle's translation vector and rotation vector to the calculated ones. In this way, all particles which survive from the evaluation procedures will represent a real sphere in the scene.



(a) (b)

Fig. 6. The definition of the object coordinate in two configurations. (a) In configuration I, the origin of the object coordinate is defined as the center of the sphere and its x-y plane is parallel to the plane formed by the four LEDs. (b) In configuration II, the origin of the object coordinate is defined as the center of the sphere and its x-y plane is parallel to the cardboard.

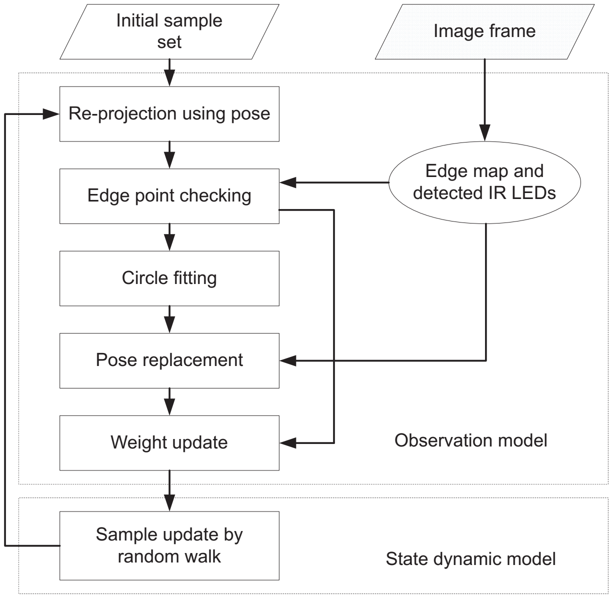


Fig. 7. The flow diagram of the proposed particle filter algorithm.

B. Configuration II: Using Encompassed Cardboard

In this configuration, we encompass the sphere in a cardboard, the center of which is coincident with that of the sphere. Such a configuration enables us to define the rotation of the sphere according to the cardboard. The object coordinate is defined in the center of the sphere, and the x-y plane of it parallels the cardboard. An illustrative figure is shown in Fig. 6 (b). The rotation of the sphere is defined as the rotation from the object coordinate to the camera coordinate. We track the sphere and the cardboard together to calculate its translation and rotation. Compared with configuration I, no Wiimote or extra camera is required. However, it inevitably causes a decrease in tracking accuracy and robustness since the Wiimote can track the IR LEDs very robustly.

1) Detection: The detection of the cardboard and sphere is combined to calculated the translation and rotation. We first use Hough transform line detection algorithm to detect the possible line segments in the initial frame and then use some simple criteria to check if four line segments form a quadrangle. The quadrangle is then used to calculate both the translation and orientation of the cardboard using the method proposed in [26]. The sphere is then projected to the image using the calculated pose. We evaluate the likelihood of the projected sphere using the method discussed in the last section. If the likelihood is above a threshold, we consider the detected pose is correct.

2) Tracking: The pose is tracked using particle filter similarly as in configuration I. The work flow and the dynamic model of the particle filter is almost the same. The difference is the observation model, i.e, how to evaluate the likelihood of the particle. We re-project the cardboard and the sphere to the image according to the pose represented by the particle, and evaluate its likelihood based on the edge map and the line segments detected by the Hough transform. Firstly, we match each side of the projected cardboard to a segment. Since some sides of the cardboard may be occluded by the sphere during the movement, not all sides can match to a segment. We discuss different cases according to the number of the segments matched.

a) 4 segments matched: The likelihood is set to the sum of two parts, the matching rate of cardboard to the matched segments, and the likelihood of the sphere. The likelihood of the sphere is calculated using the method introduced in last section. The matching rate of the cardboard is discuss in detail in [23]. If both the parts are above a threshold, that means the cardboard and sphere are matched correctly with a high confidence. We calculate the pose using the method in the detection stage, and replace the pose of the particle to it.

b) 3 or 2 segments matched: The likelihood is set similarly and the replacement is also conducted if the likelihood is above a threshold. However, the method using the quadrangle to calculate the pose is not applicable. In this case, we solve it as follows: we first calculate the translation via the fitting of the sphere. The rotation is then solved by minimizing the following back-projection errors of the corners:

$$\sum_{i=1}^{I} \|\mathbf{p}_i^{\mathrm{c}} - \hat{\mathbf{p}_i^{\mathrm{c}}}(\mathbf{K}, t_x, t_y, t_z, r_x, r_y, r_z, \mathbf{P}_i^{\mathrm{o}})\|_2^2 \quad + \\ \sum_{j=1}^{J} f(\hat{\mathbf{p}_j^{\mathrm{c}}}(\mathbf{K}, t_x, t_y, t_z, r_x, r_y, r_z, \mathbf{P}_j^{\mathrm{o}}), a_j, b_j, c_j)^2 \tag{12}$$

The first term describes the errors of the corners whose projections are the intersections of the matched segments. The second term describes the errors of the corners whose projections are not known exactly but only known to lie in the segments. The function $f(\mathbf{p}(u,v), a_j, b_j, c_j) = 0$ are the equations of the segments, i.e, $a_j u + b_j v + c_j = 0$. For the case of 3 segments matched, there are 2 corners in the first term, and 2 corners in the second term, i.e, $I = 2$ and $J = 2$. For the case of 2 segments matched, $I = 1$ and $J = 2$.

c) 1 segments matched: In this case, we simply set the likelihood to a low number and no replacement is done.

VI. Movement and View Dependant Projection

From the tracking algorithm, we know the relative pose of the sphere to the camera at each frame. We also have to know the position of the viewer's head in order to make the correct projection. Head tracking algorithms is a way to obtain the head position, but it may be not robust and accurate enough to use in our application. Alternatively, we create a fixed position where the viewer can view the projection correctly. We refer this position as the view spot. We discuss how to calibrate a view spot and generate the view and movement dependant projection.

A. View spot allocation

The allocation of the view spot is simply finding the 3D location of the view position in the camera coordinate. Our solution is to place another camera (referred as the view camera)in the view spot. By calibrating this view camera and the guide camera, we know the location and orientation of the view camera in the guide camera coordinate. The calibration is also similar. We use the sphere as the calibration object, and project a cross to the sphere surface. The cross observed by these two cameras form a correspondence. Using the estimation method before, we can calibrate the projection matrix from the guide camera to the view camera. The intrinsic parameters and the relative pose are obtained by decomposing the projection matrix. The solution are then refined by minimizing the back projection errors using LM algorithm. In this way, we allocate the view spot.

B. Projection image warping

Now, the translation $\mathbf{t}_{\mathrm{c}}^{\mathrm{o}}$ and rotation $\mathbf{R}_{\mathrm{c}}^{\mathrm{o}}$ of the object with respect to the guide camera, the relative pose from the guide camera to the view camera, denoted as $\mathbf{t}_{\mathrm{e}}^{\mathrm{c}}$, $\mathbf{R}_{\mathrm{e}}^{\mathrm{c}}$ are all obtained. We can make the movement and view dependant projection. The projection model from the object coordinate to the view camera can be given by:

$$s\tilde{\mathbf{p}^{\mathrm{e}}} = \mathbf{K}^{\mathrm{e}}(\mathbf{R}_{\mathrm{e}}^{\mathrm{c}}(\mathbf{R}_{\mathrm{c}}^{\mathrm{o}}\mathbf{V}^{\mathrm{o}} + \mathbf{t}_{\mathrm{c}}^{\mathrm{o}}) + \mathbf{t}_{\mathrm{e}}^{\mathrm{c}}) \tag{13}$$

where $\mathbf{p}^{\mathrm{e}}$ is the vertex's image in the view camera.

The next step to generate the projection image. The light path among the object, the projector, and the view camera is shown in Fig.8. An intuitive way to generate the projection image is that for each vertex of the object in object coordinate, first to find the intersection of the light path $\mathbf{V}^{\mathrm{o}}\vec{\mathbf{O}}^{\mathrm{e}}$, say P^{c}, and the sphere surface, and then project it to the projector

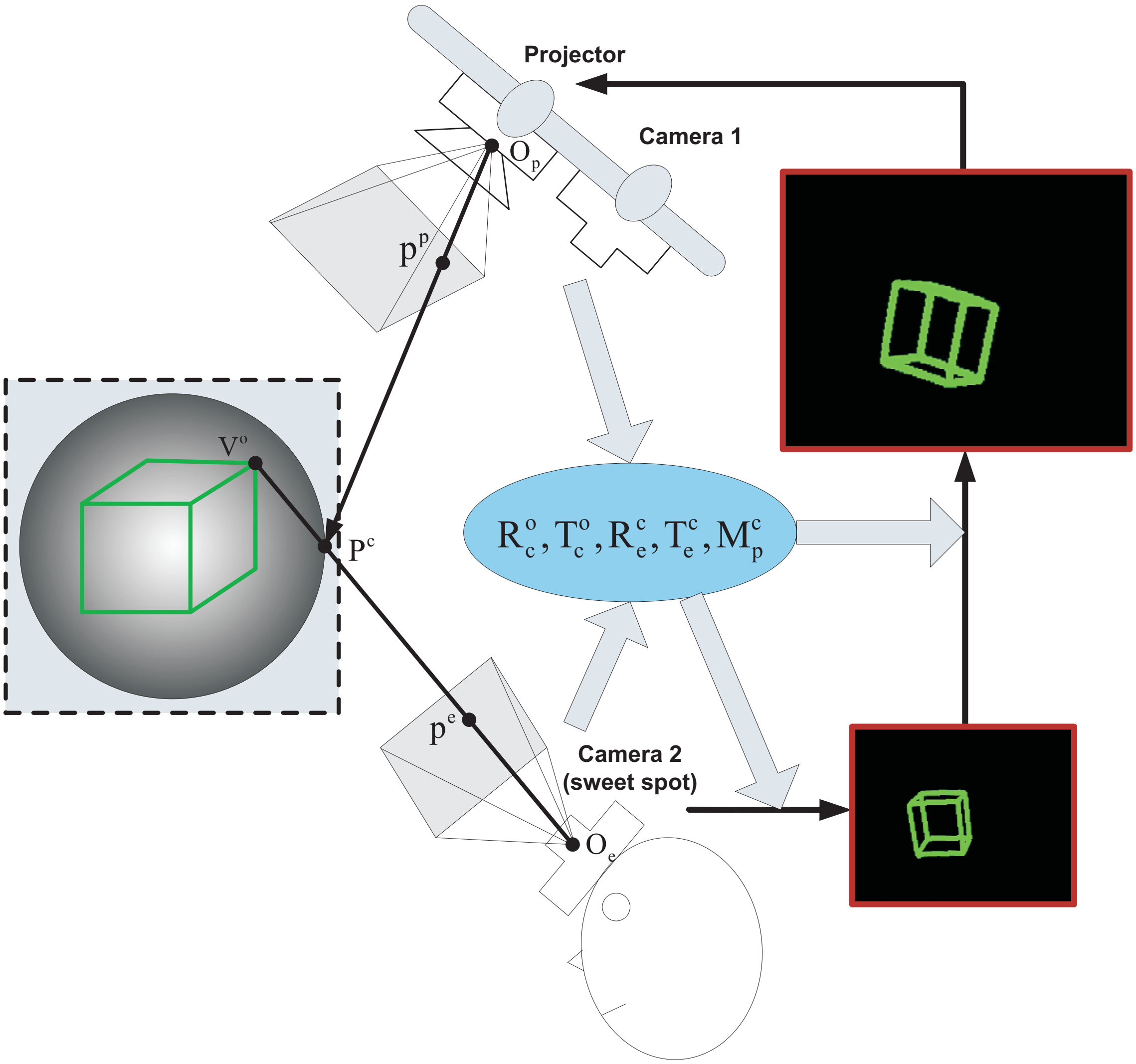


Fig. 8. Movement and view dependent projection.

image pixel$\mathbf{p}^{\mathrm{p}}$. In principle, these three points should have the same color. However, this procedure may cause some pixels of the projector image not covered, i.e, cause some holes in the projector image. To overcome this problem, we invert the procedure. We first project all vertexes of the object to the view camera using Eq. (13). Then, for each pixel in the projection image $\mathbf{p}^{\mathrm{p}}$, we find its correspondence, i.e, the correspondence point on the surface of the sphere in camera coordinate $\mathbf{P}^{\mathrm{c}}$, and the correspondence point in the view camera $\mathbf{p}^{\mathrm{e}}$. The point $\mathbf{P}^{\mathrm{c}}$ can be found by solving the following equations:

$$\begin{aligned} s\tilde{\mathbf{p}}^{\mathrm{p}} &= \mathbf{M}^{\mathrm{c}}_{\mathrm{p}}\tilde{\mathbf{P}}^{\mathrm{c}} \\ \|\mathbf{P}^{\mathrm{c}} - \mathbf{t}^{\mathrm{o}}_{\mathrm{c}}\|_2^2 &= R^2 \end{aligned} \tag{14}$$

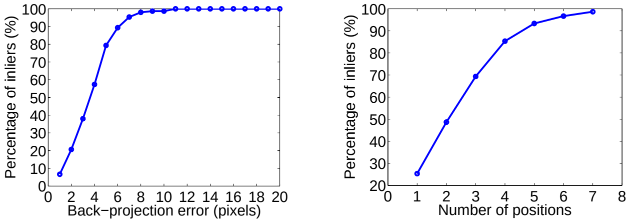
If the equations have a solution, we project it to the view camera to obtain $\mathbf{p}^{\mathrm{e}}$, and set the color of $\mathbf{p}^{\mathrm{p}}$ to that of $\mathbf{p}^{\mathrm{e}}$. Otherwise, it means $\mathbf{p}^{\mathrm{p}}$ has no correspondence point on the sphere surface. We set its pixel color to $\mathbf{0}$ in this case.

VII. Implementation And Results

We build a prototype system with the following devices: an off-the-shelf projector with resolution of 1280×1024, two Logitech Quickcam Pro 4000 webcams with resolution of 320×240 (one as the guide camera, and the other as the view camera) , a Nintendo Wiimote with resolution of 1024×768, and two foam spheres with radius of 150 mm (one for configuration I, and the other for configuration II). Four IR LEDs are embedded in a square shape on the sphere surface. The length of the arc between the diagonal LEDs is 160 mm. The size of the cardboard encompassing the sphere is 455×370 mm. A dual core 2.16GHz PC with 1GB memory is used as the testing platform. Since we are not using any special high-end devices, the cost of our system is low.

A. Projector camera pair calibration

We place the sphere in several positions to collect enough correspondences. At each position, we project the crosses evenly within the screen. Depending on the relative pose, 10$\sim$30 correspondences are collected at each position. In our implementation, we place the sphere to 7 positions, and collect totally 186 correspondences. We run RANSAC estimation 10000 iterations. The result with minimum number of outliers



(a) (b)

Fig. 9. (a) The accuracy of the projector camera calibration. (b) The accuracy vs. the number of calibration positions.

is further improved by minimizing the back-projection error. The accuracy of the estimated projection matrix is measured by the distance between the labeling points and their back-projections. We evaluate the distribution of the back-projection error, which is the percentage of the points with distance below some pixel levels (inliers) in all the labeled points. The distribution is shown in Fig. 9 (a). The mean back-projection distance is 3.6667 pixels, the standard deviation is 1.9222 pixels.

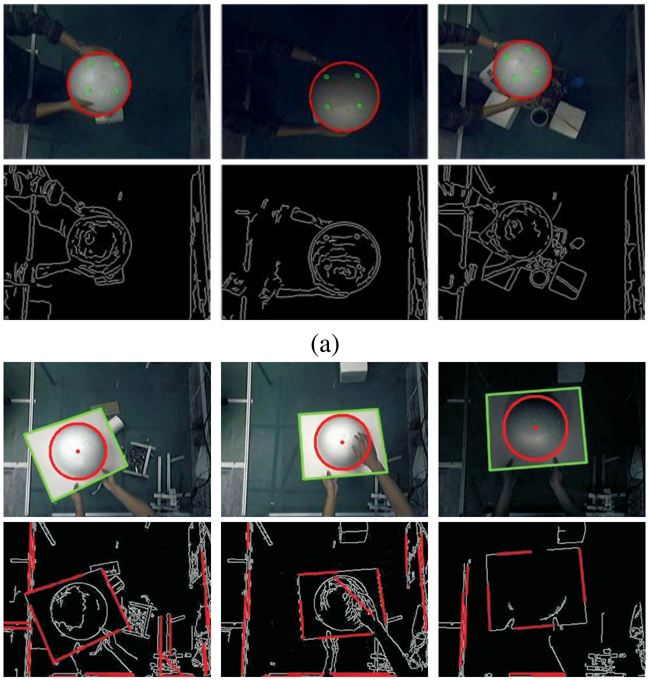
During the experiment, we found the number of positions where we placed the sphere had a significant affect to the estimation accuracy. In case of insufficient positions, the estimation result become unstable. This can be seen from the investigation of the accuracy vs. the number of positions in Fig. 9 (b). The possible reason is that the correspondences collected at one position have little difference in depth (all at the sphere surface), thus the estimation over-fits these correspondences but may not fit to the correspondences in other depths.

B. Sphere detection and tracking

Fig. 10 shows some frames extracted from the tracking process. For illustration, in configuration I, the tracked sphere is projected to the image in red and the four tracked IR LEDs in the camera are marked in green. In configuration II, the tracked sphere is also shown in red, and the cardboard is shown in green. The edge maps in both configurations are also shown.

We tested the accuracy and robustness of the trackers in tracking the translation and rotation of the sphere under different movements. A video sequence of 339 frames and 316 frames containing translation, rotation and free movements was recorded in each configuration respectively. To evaluate the tracking accuracy, We manually labeled the center and the radius of the circle, as well as the four IR LEDs (in configuration I) and the four corners of the cardboard(in configuration II). The accuracy of the circle center and the IR LEDs and cardboard corners is defined as the distance between the tracked positions of the circle and the manually labeled ground-truth. The mean and standard deviation of the tracking errors in video sequence I and II are listed in Table I. For each configuration, the first row shows the mean errors and the second row shows the standard deviation errors.

We also test the performance of the trackers under different backgrounds, including lighting changes, partial hand occlusion and dense clutter. Experiments show that both trackers can tolerate certain amount of negative factors, without significant performance loss. Our trackers may lose under extreme conditions such as over-bright and over-dark illumination, too much occlusion, round objects in the background, and too fast movements etc.



(a)

(b)

Fig. 10. Some frames of the tracking results. The circle in red is the projection of the tracked sphere. Four points in green are the tracked positions of the four IR LEDs in camera.

TABLE I
ACCURACY OF THE TRACKING I

Configuration No.	Center (pixels)	Radius (pixels)	IR LEDs or Corners (pixels)
I	3.4 1.4	4.0 2.2	6.2 3.1
II	2.8 2.1	2.9 2.3	3.6 2.1

C. Display results

We use two 3D object models to test the projection performance. One is a synthetic cube skeleton and the other is a 3D face model obtained from the USF Human ID 3-D database[27]. We test them with different types of motions of the sphere, including pure translation, pure rotation, and free movement. Fig. 11 shows some result frames of the cube in free movement. The first row shows the projection images and the second row shows the corresponding images captured by the camera. Fig. 12 shows some result frames of the face. The display results in configuration II are shown in Fig. 13 and Fig. 14. All these results show that the images can be warped and projected on the sphere precisely to create the desired effect. When examining the floor region under the sphere, we find that there is nearly no projection light on it. This means

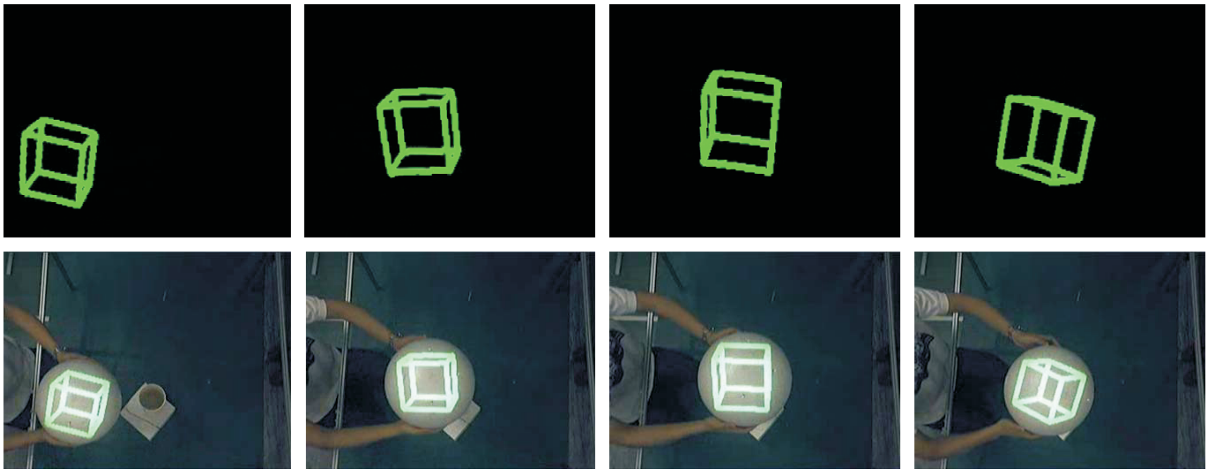


Fig. 11. Some frames of the projection results of a cube skeleton. The first row shows the generated projection images and the second row shows the corresponding images captured by the camera.

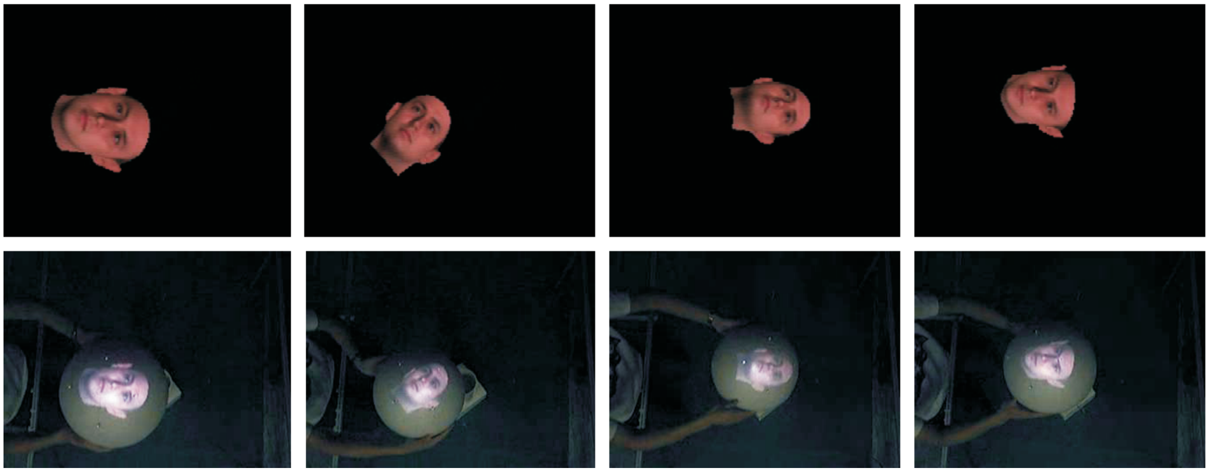


Fig. 12. Some frames of the projection results of a 3D face model.

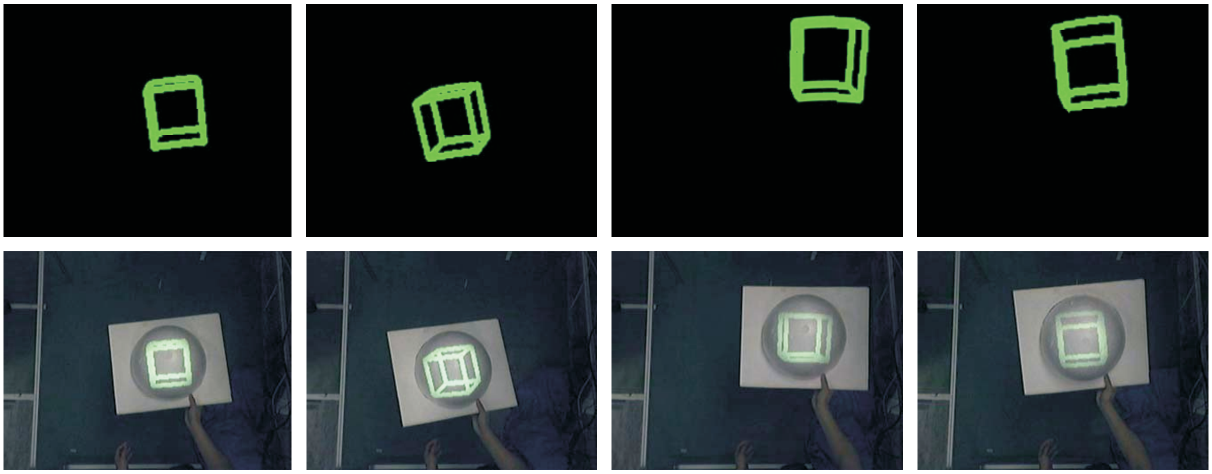


Fig. 13. Some frames of the projection results of a cube skeleton. The first row shows the generated projection images and the second row shows the corresponding images captured by the camera.

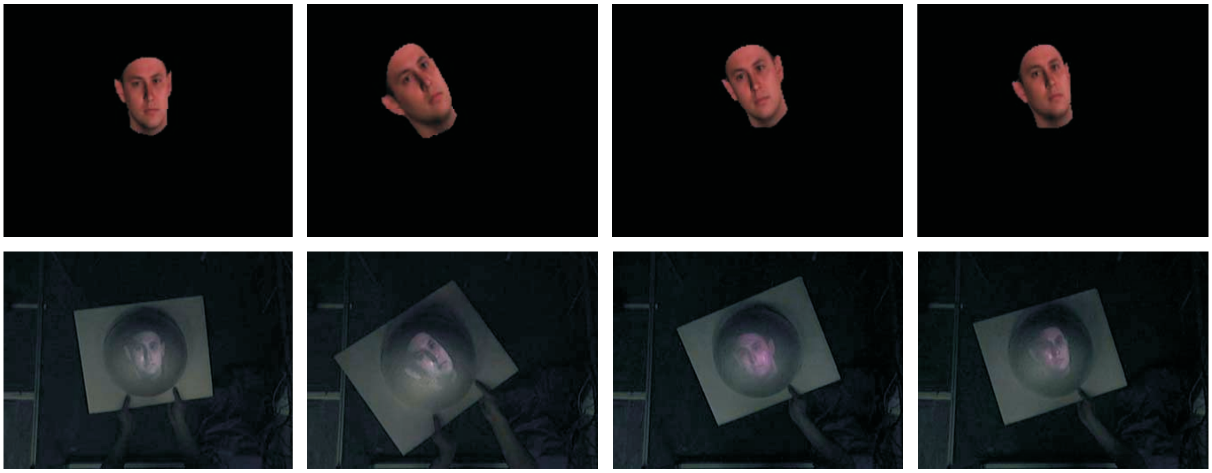


Fig. 14. Some frames of the projection results of a 3D face model.

that the image is projected to the sphere without mismatch. In all of our experiments, our system can track the sphere and generate the projection image with satisfactory accuracy and robustness. More results can be found in the supplementary video.

D. Performance

In the 2.1GHz CPU and 1GB memory platform, our sytem can achieve real-time processing (about 20 fps) smoothly in both configuration. The configuration II is slower than the configuration I because it is necessary to evaluate the likelihood of both the sphere and the cardboard. The running time mainly distributes in the edge and line feature detection, the particle filter tracking and the projection image warping. Table II shows the partition of the running time. The particle filter tracking consumes the major part of the time. It varies with the number of particles used. Fig. 15 shows the processing time against the number of particles in configuration II. The processing time increases linearly with the number of particles. In our system, the number of particles is set to 80 and 60 in the two configurations respectively. The number of line features is also an influence factor of the processing time in configuration II. We fix it to 20 in our experiments.

TABLE II
RUNNING TIME PER FRAME I

Process	Configuration I	Configuration II
Edge and line detection	about 5 ms	about 15 ms
Particle filter tracking	25 ∼ 50 ms	30 ∼ 60 ms
Projection image warping	about 5 ms	about 5 ms

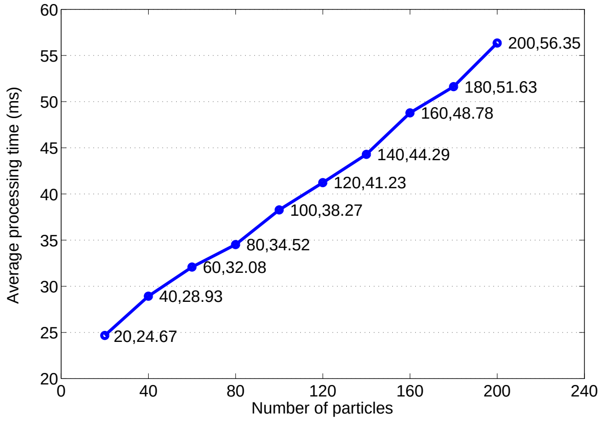


Fig. 15. The processing time of the sphere and cardboard tracking algorithm in configuration II against the number of particles.

E. Limitations

There are several limitations of our system. First, there is limitation on the resolution of the projection. Since the resolution of the projection image depends on the distance between the sphere and the projector, the projection image is inevitably downsampled when the sphere is further away from the projector. Such limitation make the small details of the displaying object unobservable or blurred. High resolution projector is a simply solution to this problem. With the reducing prices of high resolution projectors in recent years, we believe that the limitation can be overcome easily. Second, the depth field of the projector is another problem. We use a single projector in our system. The depth of filed is quite limited, making the projection in focus only within a particular range of depth. When the sphere become bigger, some parts of the sphere may become blurred. One solution to this problem is to use multiple projectors. Third, the tracking robustness and the processing time may also be limitations of our system. The unstable tracking will cause the projection results shaking, and the processing time may cause apparent latency to the projection results, especially when the sphere is moved quickly. Increasing the number of particles can improve the robustness of the tracker, but more particles mean more processing time. A tradeoff between the tracking robustness and the processing time should be made. The latency is also due to the physical latency of the projector and the camera.

High quality projector and camera can be a solution to reduce reduce the latency.

VIII. Conclusion

We have proposed an interactive projector-camera based 3D Model exhibition system using low-cost devices and computer vision techniques. The particle filter technique and a commercially available tracking product (Wiimote) are used to track the translation and rotation of the sphere respectively. The generation of the projection image is based on the translation and rotation of the sphere as well as the pre-calibrated projective geometry. Extensive experiments show that our system can robustly track the movement of the sphere and correctly generate the projection image. It successfully creates the effect with satisfactory accuracy and robustness in different lighting environments. Future work will be done to improve the accuracy, robustness and interactiveness of the system, including improving the motion tracker, and developing an algorithm to track the position of the viewer's head etc.

Acknowledgment

The authors would like to thank...

References

[1] C. Cruz-Neira, D. J. Sandin, and T. A. Defanti, "Surround-screen projection-based virtual reality: the design and implementation of the cave," in *Proceedings of ACM SIGGRAPH*, 1993, pp. 135–142.

[2] S. J. Gibbs, C. Arapis, and C. J. Breiteneder, "Teleport – towards immersive copresence," *Multimedia System*, vol. 7, no. 3, pp. 214–221, 1999.

[3] O. Bimber, G. Wetzstein, A. Emmerling, and C. Nitschke, "Enabling view-depedent stereoscopic projection in real environments," in *IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2005, pp. 14–23.

[4] R. Raskar, K. Low, and G. Welch, "Shader lamps: Animating real objects with image based illumination," *Techinal report, Chapel Hill, NC, USA*, 2000.

[5] K. L. Low, G. Welch, A. Lastra, and H. Fuchs, "Life-sized projector-based dioramas," in *ACM symposium on Virtual reality software and technology*, 2001, pp. 91–101.

[6] M. Grossberg, H. Peri, S. Nayar, and P. Belhumeur, "Making one object look like another: controllling appearance using a projector-camera system," in *CVPR*, 2004.

[7] T. Amano and H. Kato, "Appearance enhancement using a projector-camera feedback system," in *ICPR*, 2008.

[8] R. Sukthankar, R. G. Stockton, and M. D. Mullin, "Smarter presentations: Exploiting homography in camera-projector systems," in *Proceedings of ICCV*, 2001.

[9] R. Raskar and P. Beardsley, "A self-correcting projector," in *CVPR*, 2008, pp. 504–508.

[10] T. Okatani and K. Deguchi, "Autocalibration of a projector-camera system," *IEEE Trans. PAMI*, 2005.

[11] A. Raij and M. Pollefeys, "Auto-calibration of multi-projector display walls," in *ICPR*, 2004, pp. 14–17.

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

[13] O. Bimber, D. Iwai, G. Wetzstein, and A. Grundhofer, "The visual computing of projector-camera systems," in *SIGGRAPH 2008*, 2008, pp. 1–25.

[14] R. Raskar, M. S. Brown, R. G. Yang, W. C. Chen, G. Welch, H. Towles, W. B. Seales, and H. Fuchs, "Multi-projector displays using camera-based registration," in *IEEE Visualization*, 1999, pp. 161–168.

[15] R. Raskar and van B. Jeroen, "Low-cost multi-projector curved screen displays," in *International Symposium Society for Information Display (SID)*, 2005.

[16] D. Kondoh and R. Kijima, "Proposal of a free form projection display using the principle of duality rendering," in *Proceedings of VSMM*, 2002, pp. 346–352.

[17] D. Kondo, R. Kijima, and Y. Takahashi, "Dynamic anatomical model for medical education using free form projection display," in *Intl Conf. on Virtual System and Multimedia*, 2007, pp. 346–352.

[18] J. C. Lee, S. E. Hudson, and E. Tse, "Foldable interactive displays," in *ACM symposium on User Interface Software and Technology*, 2008, pp. 287–290.

[19] D. Bandyopadhyay, R. Raskar, and H. Fuchs, "Dynamic shader lamps: Painting on movable objects," in *In Proceedings of Int. Symp. On Augmented Reality*, 2001, pp. 207–216.

[20] J. C. Lee, S. E. Hudson, J. W. Summer, and P. H. Dietz, "Moveable interactive projected displays using projector based tracking," in *Proceedings of the ACM Symposium on User Interface Software and Technology*.

[21] S. Borkowski, O. Riff, and J. L. Crowley, "Projecting rectified images in an augmented environment," in *International workshop on Projector Camera System*, 2003.

[22] S. Gupta and C. Jaynes, "Active pursuit tracking in a projector-camera system with application to augmented reality," in *CVPR workshop*, 2005.

[23] K. H. W. M. C. Leung, K. K. Lee and M. M. Y. Chang, "A projector based movable hand-held display system," in *CVPR*, 2009.

[24] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[25] K. Y. Wong, D. Schnieders, and S. Li, "Recovering light directions and camera poses from a single sphere," in *Proceedings of ECCV*, 2008, pp. 631–642.

[26] Z. Y. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.

[27] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d-faces," in *Proceedings of ACM SIGGRAPH*, 1999, pp. 187–194.

PLACE
PHOTO
HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.