- **Zhaorong Li , Kin-Hong Wong, Yibo Gong, and Michael Ming-Yuen Chang, "A low-cost projector-based hand-held flexible display system", Multimedia Tools and Applications, Springer (2011): 1-20, January 14, 2011 (draft)**

A low-cost projector-based hand-held flexible display system

Zhaorong LI[1] , Kin-Hong Wong[1], Yibo Gong[1], and Michael Ming-Yuen Chang[2]

[1]*Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong*

{zrli,khwong,ybgong}@cse.cuhk.edu.hk

[2]*Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong*

mchang@ie.cuhk.edu.hk

Abstract

In this paper, we propose a low-cost hand-held flexible display system which uses a projector to project display content onto an ordinary white paper that can be twisted freely. The ultimate goal is to develop an interactive viewing tool for displaying content on flexible surface that can be deformed by the user, i.e., when the user twists the paper, the display content on the paper deforms simultaneously. Such a system may have a lot of potential in the entertainment and education industries. A pair of cameras is employed to track the pattern printed on the back of paper. They and the projector are calibrated off-line via a simple and convenient method. A real-time algorithm is proposed to recover the 3D surface of the paper. The display content is then pre-warped according to the recovered surface and projected onto the front of the paper. Two demonstrative applications are elaborated to illustrate the potential of the proposed system. Our system is easy to set up and runs in real-time. Experimental results show that the flexible display is created with satisfactory accuracy and robustness.

Keywords

Flexible display, interactive display system, multi-view reconstruction, 3D non-rigid surface tracking

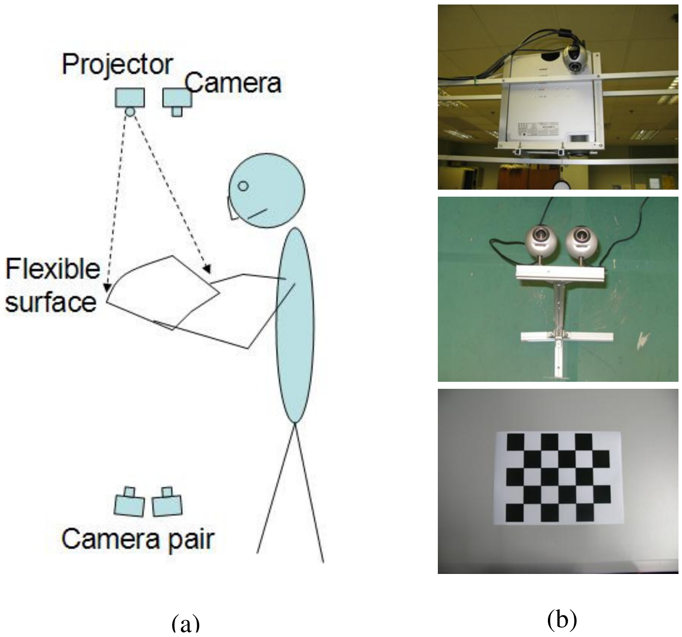
1 Introduction

In this paper, we present a flexible display system which employs a projector to project display information onto a hand-held flexible surface (e.g. an ordinary white paper) that can be twisted freely. While the user twists the projection surface, the system projects well-tailored information onto the surface corresponding to the deformation so that the viewer sees the information as if it is printed on the paper.

Traditional display systems usually display information on static flat monitors and the viewer interacts with it using indirect pointing devices such as keyboard and mouse. The shape of the screen is fixed and the control of the display such as the viewing angle is limited. As display technology being widely used in different disciplines, static display technology is not sufficient for many emerging applications. For example, in the medical field, a common way for clinicians to analyze medical volumetric data such as MRI and CT is to view the cross-sectional slices of the data obtained. With a traditional static display system, the slices can only be displayed on a fixed screen and the interaction is achieved via keyboard and mouse; the user's viewing experience is limited and the interaction is unnatural. An alternative way is to interact with these cross-sectional slices directly using a projector and a hand-held screen. The projection frustum forms a virtual object model in space, the user inserts the portable screen into the frustum and the corresponding cross-sectional slice of the volumetric data is displayed. The user can observe any cross-sectional slices he wants. This mobile type of display provides the user with an immersive experience and a more natural and direct way of interaction. Also the human internal organs are usually not planar but curved, so a flexible slicing tool that not only displays planar slices but also curved slices may help the doctor diagnose the disease. Though several flexible display systems like [5][6] have been proposed, they have limitation of requiring sensors attached onto the surface and only allow very limited deformation. These reasons motivate us to develop a more flexible, low cost, easy-to-setup and real time hand-held display system. Such a system is expected to have a lot of potential in practice, not only in medical fields but also in entertainment and education. For example, it can be used as a model-preview tool in manufacturing to preview the appearance of flexible models and how it can be twisted by users before it is put into production.

The proposed system is based on computer vision technology and the devices used consist of a projector and three webcams. Neither special hardware nor sensors are needed. The configuration of our system is shown in Figure 1. The projector and one webcam are fixed on a rig and another two webcams are placed on the floor. An ordinary white paper with printed checker pattern on the back is used as the projection screen.

When the user manipulates the paper screen within the viewing field of the projector and the cameras, the camera pair on the floor tracks the checker pattern and the projector will project pre-warped images onto the top of the paper. The facing down camera on the top is included as an observation camera when calibrating the projector since the facing up cameras on the floor cannot observe the projection result. If we reuse one of the tracking cameras as the observation camera, the total number of cameras needed is actually two.



(a) (b)

Figure 1. The configuration of our system: (a) The overall diagram of our system; (b) From top to down: the facing down projector and camera, the facing up tracking camera pair, and an ordinary paper with checker pattern printed on the back as the flexible surface.

The potential use of the proposed system is demonstrated with two applications. The first one is the flexible image projection application that can be used as a model previewing tool to view different appearances of curved model surfaces. The second one is a flexible slicing tool that not only views planar but also curvilinear cross-sections of medical volumetric data. The use of the system is not limited to these two examples. Other applications can be implemented using the similar method depending on one's imagination.

The major contribution of this work is the proposal of a new flexible display system and an effective approach to realize it with several low-cost and off-the-shelf devices. The advantage of the proposed method mainly lies in a simple calibration technique and an efficient algorithm to track and recover the deformation of the flexible surface. The rest of the paper is organized as follows: we first briefly review some related work in Section 2 and then give an overview of the system architecture in Section 3. In Section 4 we introduce the calibration of the system. In Section 5 we describe how to track and recover the surface of the paper. In Section 6 we present some potential applications of our system. Implementation and experimental results are detailed in Section 7. We conclude the paper in Section 8.

2 Related work

There are a large number of projector camera systems proposed in the literature. The problem of non-rigid surface recovery is also a widely-studied topic. In this section, we review several related projection systems and recovery algorithms.

2.1 Projector camera system

Research of building projector camera systems is popular in Human Computer Interaction (HCI) since such systems provide easy ways of man machine interaction. In most existing systems, the screens are static and non-deformable. For example, Sukthankar et al. [12] proposed a smart presentation system to correct the keystone distortion caused by the arbitrary projector-screen geometry. Raskar et al. [9] proposed a spherical screen to create an immersive large display. Although movable display systems such as [13][7] allow more freedom in control, they still project on rigid projection surfaces.

As one of the few flexible screen systems, Lee et al. [6] proposed to display content on some regularly-foldable surfaces such as scroll, fan and umbrella. They attached IR LED markers on the control points of the surface and tracked them using a Wiimote. The surface is then recovered via the markers. Since the Wiimote can track at most 4 IR LEDs at the same time, the foldable surfaces are limited to those having fairly high folding symmetry. Konieczny et al. [5] built a flexible projection system to project display content onto a flexible surface. However, this approach uses position sensors to track the surfaces. Moreover, it only allows the surface to bend in one dimension. Our approach goes beyond that in two aspects: (1) we do not need specially designed surface or sensors. Computer vision technology is used to track and recover the surface of an ordinary white paper; (2) we allow more freedom in the deformation of the surface. The user can twist the paper freely.

2.2 Non-rigid surface recovery

The non-rigid surface recovery problem refers to estimating the 3D shape of the surface based on its 2D image observation. It is a severely ill-posed problem in the case of a monocular camera since the depth information is lost under perspective projection. Many prior models and regularization methods have been proposed to solve the ill-posedness. For example, Bregler et al. [2] proposed a factorization method to build the 3D model from the tracked 2D feature points. They simply represented the 3D model as a linear combination of a set of basis vectors, which may limit the deformation ability of the model. Physics based methods [3] have better approximation to model the behavior of a general surface, but the complexity of the model may be very high. Statistical learning techniques [14][11] are usually employed to simplify complex nonlinear models. However they need a lot of training data in order to obtain a good model.

Recently Salzmann *et al.* [10] proposed a tracking method for 3D surface recovery based on a simple triangulation model. They proposed to constrain the edge orientations of the triangulation model between consecutive frames and formulated it as a Second Order Cone Programming (SOCP) problem. Though state of art result can be obtained, the computation time is very long, about 10 seconds is needed to process a frame. In order to achieve real time recovery, we use a stereo camera pair to solve the depth ambiguity. A flexible triangulation model which allows more freedom of deformation is also proposed.

3 System overviews

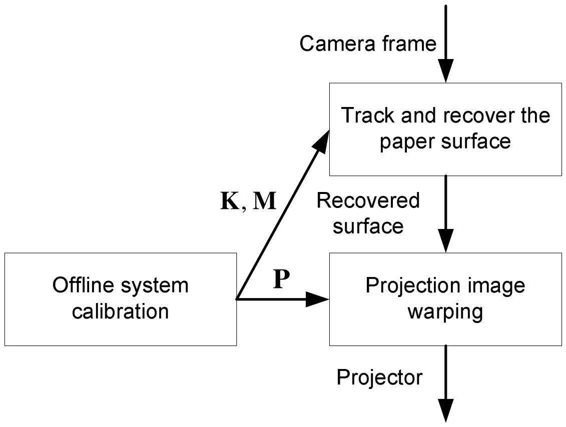


Figure 2. The overview of our system

Our system is an integration of three parts. Figure 2 shows the overview of the system. The first part is the calibration module. In order to recover the flexible surface and guide the projection to fit the deformation, we need to calibrate the geometric relationships among the two tracking cameras and the projector. In our approach, it is unnecessary to explicitly estimate the relative poses between them. Instead we simply estimate two projection matrices. A simple and convenient calibration method is proposed. The second part is tracking and recovering the surface of the paper. To simplify the tracking process and enable real time recovery, a checker pattern is printed on the back of the paper and a stereo camera pair is used. We track corners of the checker pattern and recover their 3D positions. The surface of the paper is then approximated by a triangulated mesh of the 3D corners. We employ a flexible triangulation model that can enhance the deformation ability of a fixed triangulation model. The last part is to project the display content onto the paper. Based on the calibrated projection matrix and the recovery result, the display content is pre-warped to fit the surface of the paper. In Section 4, 5, and 6, we describe each module in detail.

4 System calibrations

The calibration step finds the geometric relationships among the two tracking cameras and the projector. In our approach, we first calibrate one tracking camera using the OpenCV toolbox [4] and choose it as a reference camera. Then we calibrate two geometric relationships, one between the reference camera and the projector and the other between the two tracking cameras. Without the need to known explicit geometric parameters, we simply estimate two projection matrices, one from the 3D camera coordinate of the reference camera to the projector image plane, and another to the second tracking camera image plane. Both projection matrices are constant and independent from the deformation of the paper. While designing the calibration method, we keep in mind that the calibration process should be as easy as possible since the layout of the cameras and the projector might change frequently in practice. It should not take the user too much time and labour to calibrate the system.

4.1 The projective model

The projective model of the 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 related by a 3×4 perspective projection matrix. We assume that the world coordinate is identical to the reference camera coordinate in this paper. Then any 3D point in the reference camera coordinate, for example a point $\mathbf{X}(x, y, z)$ on the paper, corresponds to its projector pixel $\mathbf{x}(u, v)$ via a projection matrix $\mathbf{P}$:

$$s\tilde{\mathbf{x}} = \mathbf{P}\tilde{\mathbf{X}} \tag{1}$$

and

$$\mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{34} & p_{32} & p_{33} & p_{34} \end{pmatrix} \tag{2}$$

where $\tilde{\mathbf{x}}$, $\tilde{\mathbf{X}}$ are the homogenous coordinates and s is a scale factor.

Similarly, the 3D point $\mathbf{X}$ in the reference camera coordinate, and its projection $\mathbf{y}$ in the second tracking camera are also related via a 3×4 projection matrix $\mathbf{M}$:

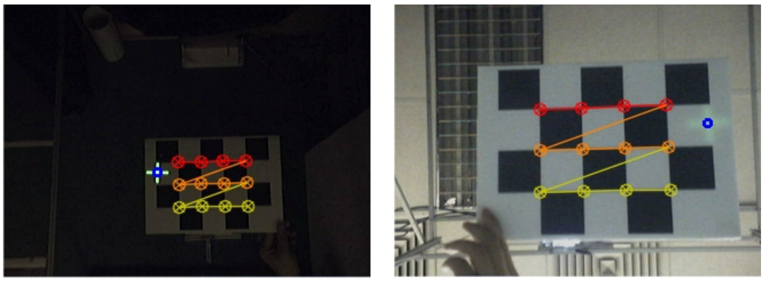
$$s\tilde{\mathbf{y}} = \mathbf{M}\tilde{\mathbf{X}} \tag{3}$$

and

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{34} & m_{32} & m_{33} & m_{34} \end{pmatrix} \tag{4}$$

The target of the calibration is then subject to estimating the two projection matrices $\mathbf{P}$ and $\mathbf{M}$.

4.2 Calibration method



(a) (b)

Figure 3. The corresponding images captured by (a) the observation camera and (b) the reference camera. The cross in the reference camera found via the homography is shown in (b).

To estimate the projection matrix $\mathbf{P}$, the main idea of the proposed method is to collect a number of correspondences between the 3D points in the reference camera coordinate and their 2D projections in the projector image. The collecting process is conducted as follows: we hold a thin cardboard with identical checker pattern printed on both sides between the projector and the reference camera. A cross with a known position is projected to the top side of the cardboard. The reference camera and the observation camera can observe the checker pattern on each side but only the observation camera can observe the cross. When the user moves the cardboard slowly, our calibration program will detect the checker pattern in both cameras and the cross in the observation camera automatically. If two checker patterns and the cross are all detected, the program reports a correspondence and asks the user if it is acceptable. This allows the user to discard bad correspondences since the detection of the checkers and cross may be wrong. If the user accepts it, our program will record the image positions of the detected checkers and cross. An example correspondence is shown in Figure 3. We repeat the above process to collect a number of correspondences. For each correspondence, the projector image position of the cross is predefined. What leaves for us to solve is its 3D position in the reference camera coordinate.

Based on the recorded image positions of the checker corners in the reference camera, we can compute their 3D coordinates by a pose estimation algorithm [15]. Accordingly, it is possible to calculate the 3D coordinate of the cross on the cardboard given its 2D image position. However, the facing up reference camera cannot observe the cross directly. So we need to find its image position in the reference camera. Since the checker patterns on the two sides are identical, they are assumed to have the same 3D positions and thus the checker corners in the two cameras are related by a homography:

$$s\tilde{\mathbf{x}} = \mathbf{H}\tilde{\mathbf{y}} \tag{5}$$

and

$$\mathbf{H} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \tag{6}$$

where $\mathbf{x}(u,v)$ and $\mathbf{y}(\alpha,\beta)$ are the corresponding corners in the reference camera and observation camera respectively. The homography matrix has 8 unknowns (up to a scale factor) and four corresponding corners are enough to estimate it. Substituting each pair of corresponding checker corners into (5) and re-arranging it to the form in (7), we can estimate the homography by Singular Value Decomposition (SVD). The image position of the cross in the reference camera is then calculated via the homography given the detected image position of the cross in the observation camera.

$$\begin{pmatrix} \alpha & \beta & 1 & 0 & 0 & 0 & -u\alpha & -u\beta & -u \\ 0 & 0 & 0 & \alpha & \beta & 1 & -v\alpha & -v\beta & -v \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix} = 0 \tag{7}$$

Next we calculate the 3D coordinates of the cross in the reference camera coordinate. We first calculate the 3D positions of the checker corners via the pose estimation algorithm [15] given the intrinsic parameters of the reference camera and the physical width of the checker. After obtaining the 3D positions of the checker corners, we can construct the planar equation of the cardboard:

$$ax + by + cz + d = 0 \tag{8}$$

where a, b, c, d are the coefficients of the planar equation. Since the 3D cross point is on the cardboard, it should satisfy the planar equation (8). Meanwhile, the 3D cross point and its 2D projection should satisfy the projection equation of the reference camera:

$$s\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K}\begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{9}$$

and

$$\mathbf{K} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \tag{10}$$

where $\mathbf{K}$ is the calibrated intrinsic parameter matrix of the reference camera and (u, v) have been obtained via the homography transformation in (5). From (8) and (9), we can solve the 3D coordinates of the cross point. Until now, we have obtained full information from feature correspondences to estimate the matrix $\mathbf{P}$.

The projection matrix in Eq. (4) has 12 unknowns (up to a scale factor), so a minimum number of 6 correspondences are enough to solve it. Substituting each pair of the 2D and 3D coordinates of the cross into Eq. (1), we can obtain a solution using SVD in the same way we estimate the homography. In order to compensate for the detection errors of the cross and the checkers, a fine adjustment is carried out. It minimizes the following sum of squared back-projection errors:

$$\sum_i \left(u_i - \frac{p_{11}x_i + p_{12}y_i + p_{13}z_i + p_{14}}{p_{31}x_i + p_{32}y_i + p_{33}z_i + p_{34}} \right)^2 + \left(v_i - \frac{p_{21}x_i + p_{22}y_i + p_{23}z_i + p_{24}}{p_{31}x_i + p_{32}y_i + p_{33}z_i + p_{34}} \right)^2 \tag{11}$$

Taking the SVD solution as initialization, we use the Levenberg-Marquardt method [8] to minimize the error. After this step, the accuracy of the estimated projection matrix is further improved.

The calibration of the projection matrix $\mathbf{M}$ can be done in a similar way. Each corner pair of the checker pattern in the two tracking cameras forms a correspondence. The calibration process is thus easier since the correspondence can be directly observed.

The proposed calibration approach is easy, flexible and automatic. The whole process involves little labour of the user. It takes about a few minutes to complete a whole calibration, including collecting the correspondences and estimating the two projection matrices.

5 Paper Surface tracking and Recovery

We propose a real-time surface tracking and recovery algorithm to recover the 3D surface of the paper in each frame. In our implementation, to simplify the tracking and save computation time, a checker pattern is printed on the back of the paper, which can be easily detected and tracked by the Lucas-Kanade tracker [1]. Moreover, the calibrated tracking camera pair is used to solve the depth ambiguity. The recovery task is subject to recovering the 3D positions of the checker corners in each frame based on their tracked image positions. Assuming there are totally n corners, and the tracked positions in the two tracking cameras are $x_i(u_i, v_i), y_i(\alpha_i, \beta_i), i = 1 \cdots n$, the corresponding 3D positions $\mathbf{X}_i(x_i, y_i, z_i)$ are then the unknown variables to be estimated.

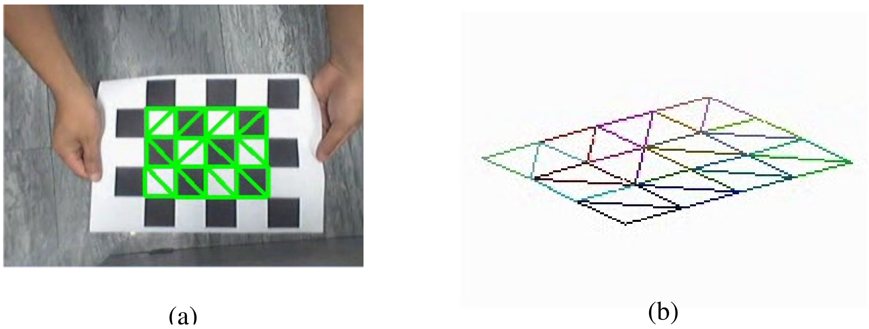


Figure 4. An example of the flexible triangulation of a 4×3 checker pattern. The triangulation in (b) is re-projected to the paper in (a).

To allow more deformation freedom, we apply a flexible triangulation scheme to the corners. In detail, each checker is triangulated through introducing a diagonal line. However, different from existing fixed triangulation models, the choice of which diagonal line to triangulate along is not fixed beforehand but to be determined during the recovery process. In other words, the model allows each checker to deform along either one of its two diagonal lines. The flexible triangulation model enhances the deformation ability of the fixed triangulation model, especially for our small size checker pattern. We introduce a variable ω for each checker to indicate along which diagonal to triangulate. The variable has two possible values, 1 or -1. The value 1 indicates the checker deforming along the left-top to right-bottom diagonal line while -1 indicates the deformation along the other diagonal line. Supposing there are m checkers in the pattern, the indication variables $\omega_i, i = 1 \cdots m$ are also the unknowns to be estimated. An example of a flexible triangulation to a 4×3 checker pattern is shown in Figure 4.

To solve the 3D positions and the indication variables, we minimize an energy function composed of two parts, the back-projection errors of the checker corners in two tracking cameras and a smoothness term to regularize the triangulation model. Substituting each corner into Eq. (9) and Eq. (3), we can formulate the energy function of back-projection errors as:

$$\mathbf{E}_p = \sum_{i=0}^{n}\left(\left(\left(\frac{\mathbf{k}_1^{\mathrm{T}}\mathbf{X}_i}{\mathbf{k}_3^{\mathrm{T}}\mathbf{X}_i} - u_i\right)^2 + \left(\frac{\mathbf{k}_2^{\mathrm{T}}\mathbf{X}_i}{\mathbf{k}_3^{\mathrm{T}}\mathbf{X}_i} - v_i\right)^2\right) + \left(\left(\frac{\mathbf{m}_1^{\mathrm{T}}\tilde{\mathbf{X}}_i}{\mathbf{m}_3^{\mathrm{T}}\tilde{\mathbf{X}}_i} - \alpha_i\right)^2 + \left(\frac{\mathbf{m}_2^{\mathrm{T}}\tilde{\mathbf{X}}_i}{\mathbf{m}_3^{\mathrm{T}}\tilde{\mathbf{X}}_i} - \beta_i\right)^2\right)\right) \tag{12}$$

where $\mathbf{k}_1^{\mathrm{T}}, \mathbf{k}_2^{\mathrm{T}}, \mathbf{k}_3^{\mathrm{T}}$ are three row vectors of $\mathbf{K}$ and $\mathbf{m}_1^{\mathrm{T}}, \mathbf{m}_2^{\mathrm{T}}, \mathbf{m}_3^{\mathrm{T}}$ are three row vectors of $\mathbf{M}$. To regularize the triangulation model, an intuitive idea is to preserve the original edge length of each triangle. However, such a regularization term is difficult to optimize. Salzmann *et al.* [10] proposed to preserve the orientation of the edge in consecutive frames and obtained good results. We employ the same constraint in our formulation and formulate it as a quadratic term. Assuming that the surface $\left\{\mathbf{X}_i^t, i = 1\ldots n\right\}$ at time t is known, for each edge $\overline{\mathbf{X}_i\mathbf{X}_j}$ in the triangulation model, the edge orientation constraint is formulated as the difference of orientation between consecutive frames, namely:

$$\delta_{ij} = \left\|\mathbf{X}_i^{t+1} - \mathbf{X}_j^{t+1} - \theta_{ij}^t\right\| \tag{13}$$

and

$$\theta_{ij}^t = L_{ij}\frac{\mathbf{X}_i^t - \mathbf{X}_j^t}{\left\|\mathbf{X}_i^t - \mathbf{X}_j^t\right\|} \tag{14}$$

where L_{ij} is the original length of the edge. According to our triangulation model, there are two types of the edges. One is the side edge of the checker and another is the diagonal line. However, for the second type, we should choose the diagonal line to regularize according to the value of the indication variable. If it is 1, we constrain the left-top to right-bottom diagonal line; If -1, we constrain the other diagonal line. The total smoothness term is the sum over all edges, namely:

$$\mathbf{E}_r = \sum_{k=1}^{m}\left(\frac{\omega_k + 1}{2}\delta_{ac}^2 + \frac{1 - \omega_k}{2}\delta_{bd}^2\right) + \sum_{(i,j)\in\mathbf{\Omega}}\delta_{ij}^2 \tag{15}$$

where a, b, c, d are indices of the four corners of the k^{th} checker, ac and bd are the two diagonals, $\mathbf{\Omega}$ is the set of side edges. The recovery is then subject to minimizing the sum of the two energy functions.

$$\min_{\mathbf{X}_i, \omega_k} \mathbf{E}_p + \lambda\mathbf{E}_r \tag{16}$$

where λ is a weight of the smoothness term. There are totally $3n + m$ variables to be solved. Simultaneous minimization over $\mathbf{X}$ and ω is difficult because the indication variables are discrete and there are enormous combinations of them even with a small size checker pattern. Our solution is to separate $\mathbf{X}$ and ω and minimize over them alternately. Specifically, when minimizing over $\mathbf{X}$, we keep ω constant, and vice visa. The two minimization phases in one iteration are detailed as follows:

minimization over $\mathbf{X}$ All $\mathbf{X}_i$ are involved in both $\mathbf{E}_p$ and $\mathbf{E}_r$. They are quadratic in $\mathbf{E}_r$ but non-quadratic in $\mathbf{E}_p$. To simplify the optimization, we reformulate $\mathbf{E}_p$ to quadratic form. The idea is to restrict the back-projection errors under a bound γ, and rewrite it to a quadratic form:

$$\begin{aligned} \left(\mathbf{k}_1^{\mathrm{T}}\mathbf{X}_i - u_i\mathbf{k}_3^{\mathrm{T}}\mathbf{X}_i\right)^2 + \left(\mathbf{k}_2^{\mathrm{T}}\mathbf{X}_i - v_i\mathbf{k}_3^{\mathrm{T}}\mathbf{X}_i\right)^2 \le \gamma^2\left(\mathbf{k}_3^{\mathrm{T}}\mathbf{X}_i\right)^2 \\ \left(\mathbf{m}_1^{\mathrm{T}}\tilde{\mathbf{X}}_i - \alpha_i\mathbf{m}_3^{\mathrm{T}}\tilde{\mathbf{X}}_i\right)^2 + \left(\mathbf{m}_2^{\mathrm{T}}\tilde{\mathbf{X}}_i - \beta_i\mathbf{m}_3^{\mathrm{T}}\tilde{\mathbf{X}}_i\right)^2 \le \gamma^2\left(\mathbf{m}_3^{\mathrm{T}}\tilde{\mathbf{X}}_i\right)^2 \end{aligned} \tag{17}$$

$\mathbf{E}_p$ then becomes:

$$\begin{aligned} \mathbf{E}_p = & \sum_{i=0}^{n}\left(\left(\mathbf{k}_1^{\mathrm{T}}\mathbf{X}_i - u_i\mathbf{k}_3^{\mathrm{T}}\mathbf{X}_i\right)^2 + \left(\mathbf{k}_2^{\mathrm{T}}\mathbf{X}_i - v_i\mathbf{k}_3^{\mathrm{T}}\mathbf{X}_i\right)^2 - \gamma^2\left(\mathbf{k}_3^{\mathrm{T}}\mathbf{X}_i\right)^2\right) \\ & + \sum_{i=0}^{n}\left(\left(\mathbf{m}_1^{\mathrm{T}}\tilde{\mathbf{X}}_i - \alpha_i\mathbf{m}_3^{\mathrm{T}}\tilde{\mathbf{X}}_i\right)^2 + \left(\mathbf{m}_2^{\mathrm{T}}\tilde{\mathbf{X}}_i - \beta_i\mathbf{m}_3^{\mathrm{T}}\tilde{\mathbf{X}}_i\right)^2 - \gamma^2\left(\mathbf{m}_3^{\mathrm{T}}\tilde{\mathbf{X}}_i\right)^2\right) \end{aligned} \tag{18}$$

and the minimization becomes:

$$\min_{\mathbf{X}_i} \mathbf{E}_p + \lambda\mathbf{E}_r \tag{19}$$

All terms in the total energy are quadratic, so the energy function can be easily minimized by solving the linear equations:

$$\frac{\partial\left(\mathbf{E}_p + \lambda\mathbf{E}_r\right)}{\partial\mathbf{X}_i} = 0 \tag{20}$$

minimization over ω The variables ω are only involved in the diagonal line of the regularization term. So we can ignore other terms. The minimization becomes:

$$\min_{\omega_k}\sum_{k=1}^{m}\left(\frac{\omega_k+1}{2}\delta_{ac}^2 + \frac{1-\omega_k}{2}\delta_{bd}^2\right) \tag{21}$$

Since each ω_k is independent, the optimization is actually a comparison of δ_{ac} and δ_{bd} for each checker. If δ_{ac} is smaller, ω_k is then set to 1. Otherwise, ω_k is set to -1.

The initial values of $\mathbf{X}$ and ω are set to the result of the previous frame. For the first frame, a tricky method is used. We require the paper in the first frame to be planar. So $\mathbf{X}$ in the first frame can be obtained simply by the pose estimation algorithm [15]. For ω, we simply set all of them to 1 in the first frame. Although the paper may not be ideally planar in practice, it works well in our experiment. The above alternate minimization converges quickly and we usually run a few iterations for each frame.

6 Applications

From the tracking and recovery algorithm, we obtain the 3D surface of the paper in each frame, represented as a triangulated mesh of 3D corners. In combination with the calibration result, we can make the flexible surface a versatile interface for visualizing images and data. In this section, we describe two demonstrative applications to illustrate the use of our proposed system.

6.1 Flexible image projection

The first one is to display a "flexible" image on the paper, i.e. when the user twists the paper, the image bends simultaneously with the paper deformation, just like it was printed on the paper. This application can be viewed as an example of shader lamp [16], in which the object to be modified is the projection surface itself. It would be useful as a model previewing tool to view different appearances of a curved surface. This kind of flexible projection can be widely used in entertaining and educational field to produce a more immersive user experience.

The display relies on a pre-warping of the display content before projected to the paper. Given the display content image $\mathcal{S}$, the warping of the projection image $\mathcal{P}$ is conducted as follows. For each triangle of the surface, we first project its three vertices to the projection image plane using the projection matrix $\mathbf{P}$. For example, a triangle composed of $\mathbf{X}_i, \mathbf{X}_j, \mathbf{X}_k$ is projected to $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$. Then for each pixel $\mathbf{x}$ within the projected triangle, we find its corresponding point $\mathbf{X}$ on the paper. We write $\mathbf{X}$ in barycentric coordinates in terms of its three vertices:

$$\mathbf{X} = \xi_1 \mathbf{X}_i + \xi_2 \mathbf{X}_j + \xi_3 \mathbf{X}_k \tag{22}$$

Since point $\mathbf{X}$ projects to the pixel $\mathbf{x}$ via the projection matrix $\mathbf{P}$, we can obtain the barycentric coordinates by solving the following linear equations:

$$\begin{aligned} s\tilde{\mathbf{x}} &= \mathbf{P}\tilde{\mathbf{X}} \\ \xi_1 + \xi_2 + \xi_3 &= 1 \end{aligned} \tag{23}$$

The target is then to set the content of $\mathcal{P}(\mathbf{x})$ to what should be display on $\mathbf{X}$. We apply the same triangulation of the surface to $\mathcal{S}$, and for each $\mathbf{x}$, with the barycentric coordinates ξ_1, ξ_2, ξ_3 calculated we can find the corresponding pixel $\mathbf{y}$ in $\mathcal{S}$. The colour of $\mathcal{S}(\mathbf{y})$ is then copied to $\mathcal{P}(\mathbf{x})$. By reversing the projection direction, the content of each triangle in $\mathcal{S}$ will be projected onto a corresponding triangle region on the surface.

A depth image $\mathcal{Z}$ with the same size of the projection image is used to handle the possible mutual occlusion of the triangles. Each $\mathcal{Z}(\mathbf{x})$ keeps the minimum depth among all the points that project to $\mathbf{x}$ in the projection image. We initialize it with a very large depth. During the warping, for each $\mathbf{X}$, if it is nearer than the depth kept in $\mathcal{Z}(\mathbf{x})$, we do the warping and replace $\mathcal{Z}(\mathbf{x})$ with the depth of $\mathbf{X}$. Otherwise, it means $\mathbf{X}$ is occluded and we simply ignore it.

Through the above warping, the display content can be shown to deform with the paper simultaneously when projecting the pre-warped images.

6.2 3D Volume Visualization

The second application is to use the flexible screen as 3D volume data visualization tool, which would be very useful for viewing cross sections of medical volumetric data such as MRI and CT. Rather than displaying the volumetric data on a fixed screen, the proposed system can be used as a slicing tool to examine the slices of the volume data in their actual positions. With this slicing tool, we can simulate a virtual volume placed at a certain position in front of the projector. When the user moves the paper within the virtual volume, the slice of the volume data is shown on the paper, just like the user is holding the actual slice of data. This would give the user a more intuitive and immersive experience, and also more freedom in interaction compared with traditional keyboard-mouse display system. Moreover, since most real volume data rarely follows a perfect plane, e.g, spine, or kidney, it is desirable that the viewer be able to view curved slices of data. With the help of our system, the observer can see details of the inner surface simply by adjusting the position and shape of the paper held by his hands. This is very useful when the doctor is analyzing the health condition of a patient or when a medical teacher is teaching the student about the structure of the human body. We believe the proposed system will have great potential in medicine and education.

The generation of the projection image is the same as the first application except for the setting of $\mathcal{P}(\mathbf{x})$ .The volume data is first loaded as a 3D texture and aligned at a certain position in front of the projector. It is also scaled to match the size of the projection screen. After that, we solve the intersection of $\mathbf{X}$ with the volume data. The intersected voxels of data are then interpolated to give the value that should be display at $\mathbf{X}$. This is also the value that should be set to $\mathcal{P}(\mathbf{x})$. Through this process, the cross section image is created and then projected onto the surface.

7 Experimental Results

We have built a prototype system with the following devices: an off-the-shelf projector with resolution of 1280×1024, and three Logitech Quickcam Pro 4000 webcams with resolution of 320×240. A dual core 2.16 GHz PC with 1 GB memory is used as the testing platform. Since we are not using any special devices, the cost of our system is low. Experimental results show that the display system achieves satisfactory accuracy and robustness with these ordinary devices.

7.1 System calibration

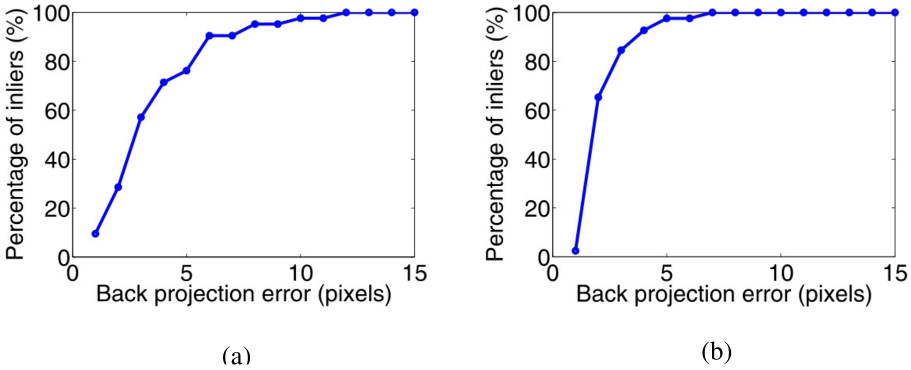


Figure 5. The distribution of the back projection error for the two calibrations: (a) Estimation of $\mathbf{P}$; (b) Estimation of $\mathbf{M}$.

We use a thin but hard cardboard with 3×2 checker pattern printed on both sides (see Figure 3) to collect correspondences. The width of each checker is 50 mm. We collect totally 48 correspondences to calibrate the projector camera pair and 72 correspondences to calibrate the tracking camera pair by changing the position and orientation of the cardboard. The whole process takes about 10 minutes. Most of the time is spent in eliminating the false detections of the cross. The calibration time can be further reduced by improving the detection. 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 distributions of the two calibrations are shown in Figure 5. The back projection error corresponding to 80% inliers for the tracking camera pair is 2.6 pixels and that for the projector camera pair is 5.3 pixels. It is an acceptable accuracy for our display application.

7.2 Paper surface recovery and tracking

The parameters of the recovery algorithm are set as follows in our experiments: the weight λ of the smoothness term is set to 1×10^5 and the back-projection error bound γ is set to 2 pixels. We run the alternate minimization between $\mathbf{X}$ and ω for 3 iterations. To evaluate the performance of the recovery algorithm, we generate a sequence of 200 synthetic surfaces by simulating a paper bending process. Some frames are shown in Figure 6. A 4×3 lattice is used and the width of each checker is 50 mm, which is the same as our checker pattern. The 3D corners are then projected to 2D with the intrinsic parameter matrix $\mathbf{K}$ of the reference camera and the calibrated projection matrix $\mathbf{M}$. Gaussian noises with standard deviation $\sigma=2$ are added to the 2D projections. We test our algorithm with fixed and flexible triangulation models on the same synthetic data. The accuracy of the surface recovery is measured in two aspects: the mean distance between the recovered corners and their ground-truth positions, and the back-projection errors. The result is shown in Figure 7. We can see that the proposed method achieves good accuracy in both mean error distance and back-projection errors, and the result with a flexible triangulation model is more accurate and stable.

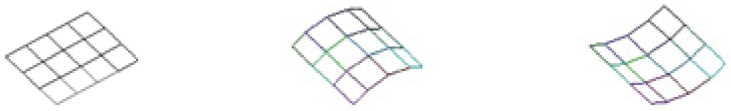


Figure 6. Some frames of the synthetic 4×3 surface sequence.

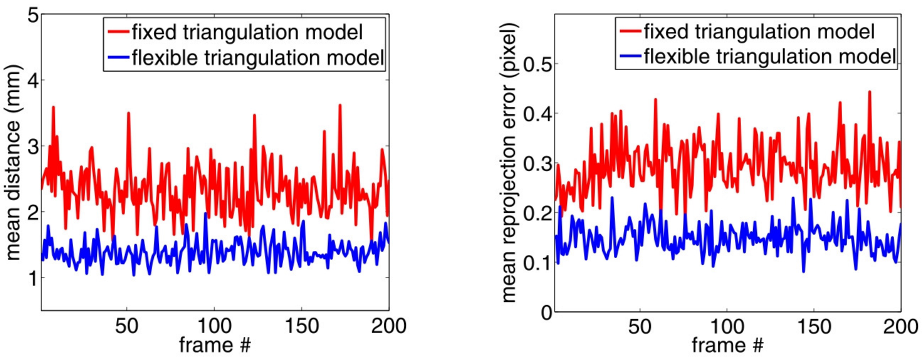


Figure 7. The performance of our algorithm on a synthetic sequence. The flexible triangulation achieves better accuracy and stability.

To evaluate the performance of our recovery algorithm in real scenarios, we test it with live webcam captures. Figure 8 shows several resulting frames of tracking and recovering a paper printed with a 4×3 checker pattern. The recovered surfaces are shown in another perspective (from the user's view). The performance of our algorithm on real data is difficult to evaluate quantitatively since the ground-truth 3D corners of the check pattern are difficult to measure. Here, we simply evaluate the recovering accuracy of the curvature of the paper since the ground-truth curvature of the paper can be measured by the height and width of the arch. We compute the curvature of the recovered surface according to the 3D corners, and then compare it with the manually measured data. Five set of deformations are evaluated and the recovering error is plotted in Figure 9. In general, our algorithm can recover the paper with a maximum curvature about 0.5 (its corresponding recovering error is about 0.05), which is enough for most of applications.

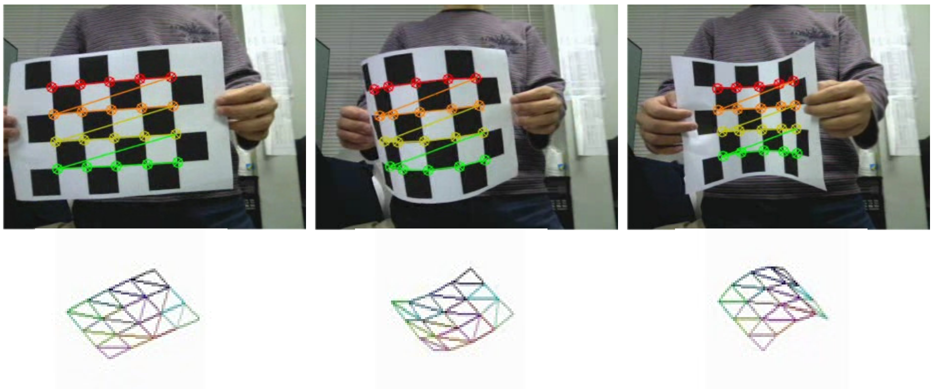
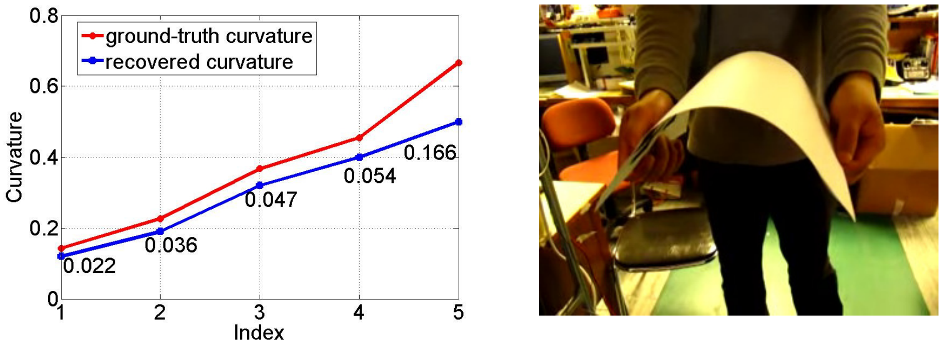


Figure 8. Recovering the surface of a paper printed with 4×3 checker pattern in a live webcam capture. The first row shows the tracked checker patterns. The second row shows the recovered surface in another perspective.



(a) (b)

Figure 9. (a) The recovering accuracy of the curvature. The errors between recovered curvature and ground-truth are also plotted. (b) The approximate maximum acceptable deformation of the paper.

The working area of the camera pair is also investigated. In general, the working area is the intersection of the field of view of the two cameras, but limited to a range in depth. To find out its size, we move the paper to everywhere it can be tracked by the camera pair. The position of the paper is estimated for each frame based on the recovered 3D paper surface and the size of the working area is then evaluated as the maximum allowed moving range of the paper. It is approximately 0.9 m in height, and 0.9 m in width at the top for our prototype configuration. The working area is not fixed, i.e., it can move in depth depending on the size of the paper. Specifically, if the paper is bigger, it will go further from the camera, and vice visa. It is thus no problem to track a bigger paper, e.g. an A3 paper. Figure 10 shows several tracking frames with an A3 paper printed with 6×4 checker pattern on the back.

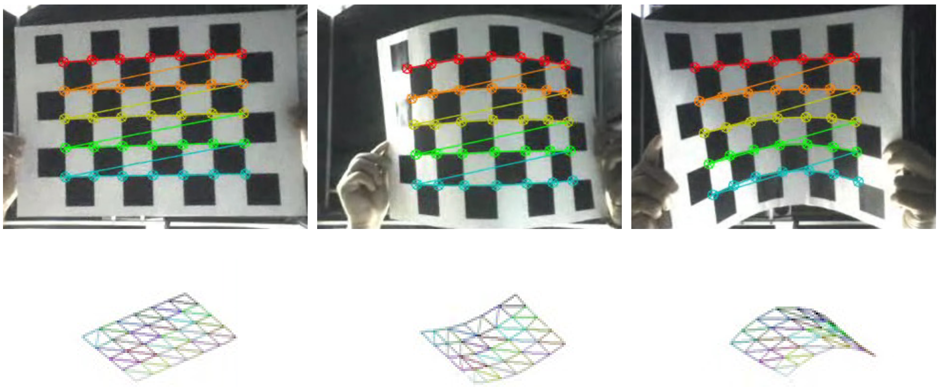


Figure 10. Tracking and recovering the surface of an A3 paper printed with 6×4 checker pattern in a live webcam capture. The first row shows the tracked checker patterns. The second row shows the recovered surface in another perspective.

7.3 Display results

To evaluate the projection accuracy, we project a green grid onto the paper on which a black grid is printed. In this case, the black grid is the ground-truth. Therefore, by observing and measuring the offset between the green grid and the black grid, we can access the projection accuracy. An evaluating video is recorded and the projection errors between the green and black grid are manually measured. The average projection error of the corner is about 2.3 pixels. Two evaluating frames are shown in Figure 11.

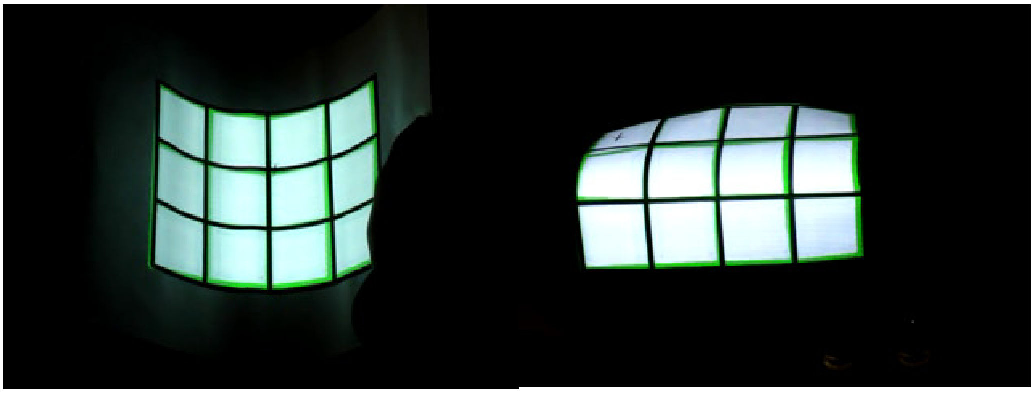


Figure 11. The green grid is projected while the black grid is printed. The coincidence of the green grid with the black grid indicates the projection accuracy of the system.

Some results of the flexible image display application are shown in Figure 12. The cooperation of the surface recovery and the image pre-warping routine can successfully project correct content on the surface under different kinds of deformations. Figure 13 illustrates some results of virtually slicing a MRI brain. From these images, it can be easily seen that a curved slice of the brain can be exhibited to the user owing to our system. In both experiments, we find that the system can warp the projection image correctly and create the flexible display with satisfactory accuracy and robustness. It runs smoothly and no obvious latency and flickering effect is observed. Continuous display results can be found in the supplementary video (http://www.cse.cuhk.edu.hk/~zrli/flexibledisplay.mpg).

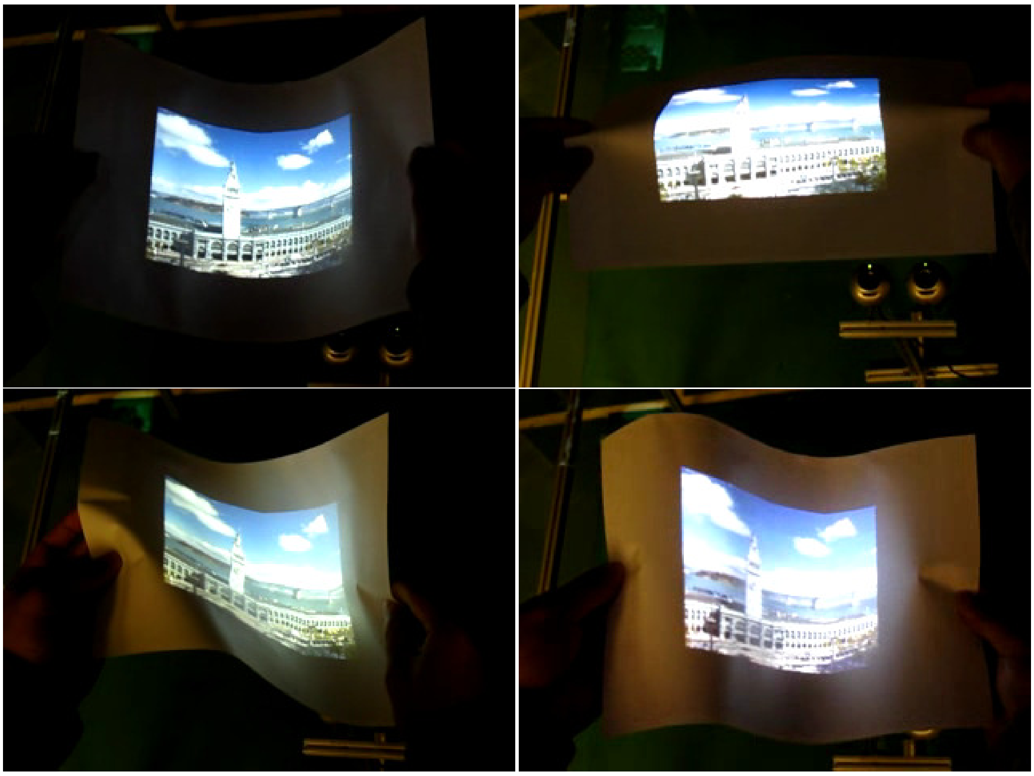


Figure 12. Some results of flexible image projection.

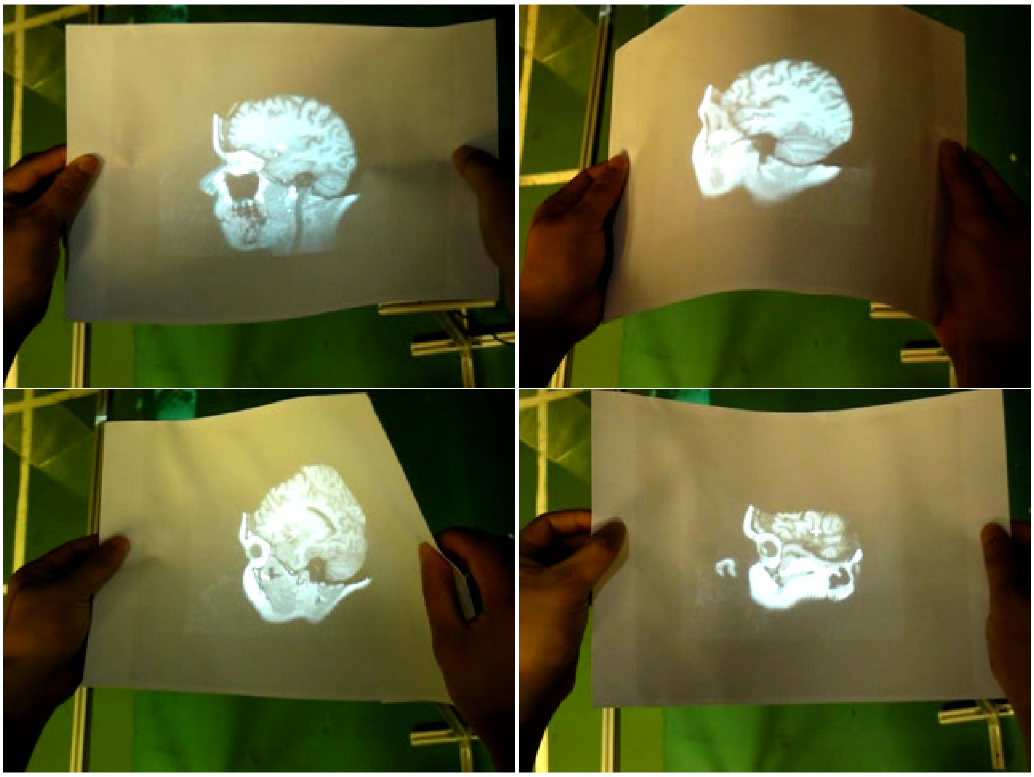


Figure 13. Some results of slicing a MRI brain.

Our system can achieve real-time performance on the above platform. The running time of each frame is mainly occupied by the checker pattern tracking (about 5 ms), the surface recovery (about 10 ms) and the projection image warping (about 15 ms). The surface recovery is actually fast while the projection image warping is slow. When running with live webcam capture, our system can still achieve real time processing smoothly (a frame rate about 18 fps). This speed is acceptable even for carrying out some fast deformations (see supplementary video). However, limited by the frame rate of the webcam and the KLT algorithm (which expects tiny variance between consecutive frames), our system cannot accurately handle drastic deformations and will exhibit perceivable latency. But if cameras with higher frame rates (like embedded cameras) can be used, this problem will be greatly relieved, and the recovery ability of the system can be further enhanced.

7.4 Discussions

Although the performance of the system is satisfactory, there are still several aspects that can be further improved. First, there is some limitation on the working area of the system, which is generally confined to the working area of the camera pair. Although it is sufficient for many visualization applications like the two demonstrated in this paper, it may be limited for some interactive applications that require large movement of the projection surface. Possible ways to enlarge the working area include using cameras with bigger field of view and focus area, or using an array of cameras to ensure the pattern can always be seen by two of them. Second, there is limitation on the depth field of the projector. The projector used in our system has quite limited depth of field, which allows the projection to be in focus only within a particular range of depth. A solution to this problem is to use multiple projectors. Third, there may be applications that require fine representation of the surface. The proposed system may fail as we are using a coarse checker pattern to approximate the surface for the benefit of easy tracking. Increasing the size of the checker pattern or using texture-abundant patterns could achieve a better approximation, but it would increase difficulty in feature tracking and matching, and also increase the running time greatly. A fast and robust tracking algorithm may be needed to make the system real time for large scale applications.

8 Conclusions

We have proposed a flexible projector-based hand-held display system using ordinary devices and computer vision technology. A projector and a camera pair are used as the projection and tracking device and an ordinary white paper is used as the flexible projection surface. No sensors or special hardware is needed in our system. An offline flexible, easy and automatic calibration method is employed to calibrate the system. A real-time tracking and recovery algorithm is proposed to track and recover the 3D surface of the paper. The display content is pre-warped and projected to the paper based on the calibration result and the recovered surface. Two model applications are elaborated to demonstrate the potential of our system. Experimental results show that our system can successfully create the flexible display on the deformable surface with satisfactory accuracy and robustness. Future work will be carried out to improve the freedom of control and interactivity of the system.

References

[1] Bouguet, J.Y. “Pyramidal implementation of the lucas kanade feature tracker”

[2] Bregler, C., Hertzmann, A., and Biermann, H, (200) “Recovering non-rigid 3d shape from image streams”. In Proceedings of intl. Conf. on Computer Vision and Pattern Recognition.

[3] Ilic, S., and Fua, P. (2006) “Implicit meshes for surface reconstruction”. IEEE *Transactions on PAMI*, Vol. 28, No. 2, pp328-333

[4] Intel. OpenCV. http://www.intel.com/technology/computing/opencv.

[5] Konieczny, J., Shimizu, C., Meyer, G. and Colucci, D. (2005) “A hand-held flexible display system”, IEEE Visualization Conference, pp591-597.

[6] Lee, J.C., Hudson, S.E., and Tse, E.(2008) “Foldable interactive displays”, In Proceedings of the 21st annual ACM symposium on User Interface Software and Technology, pp287-290

[7] Leung, M.C., Lee, K.K., Wong, K.H., and Chang, M.Y.(2009) “A projector based movable hand-held display system”. In Proceedings of intl. Conf. on Computer Vision and Pattern Recognition.

[8] Levmar. “levmar: levenberg-marquardt non-linear least squares algorithms in c/c++”, http://www.ics.forth.gr/lourakis/levmar/.

[9] Raskar, R. and Jeroen van B. (2005) “Low-cost multi-projector curved screen displays”. In International Symposium Society for Information Display (SID)

[10] Salzmann, M., Hartley, R. and Fua, P. (2007) “Convex optimization for deformable surface 3d tracking”, in Proceedings of Intl. Conf. on Computer Vision

[11] Salzmann, M., Pilet. J., Ilic, S. and Fua, P.(2007) “Suface deformation models for non-rigid 3d shape recovery”, IEEE Transactions on PAMI, Vol. 29, No. 8, pp1481-1487

[12] Sukthankar, R., Stockton, R.G., and Mullin, M.D.(2001) “Smarter presentations: Exploiting homography in camera-projector systems”, In Proceedings of Intl. Conf. on Computer Vision.

[13] Summet, J. and Sukthankar, R.(2005) “Tracking locations of moving hand-held displays using projected light”. In Proceedings of Pervasive, pp. 37-46.

[14] Xiao, J., Baker, S., Matthews, I. and Kanade, T.(2004) “Real-time combined 2d+3d active appearance models”, In Proceedings of intl. Conf. on Computer Vision and Pattern Recognition.

[15] Zhang, Z.(2000) “A flexible new technique for camera calibration”, IEEE transactions on PAMI, Vol.22, No.11, pp1330-1334.

[16] Raskar, R., Welch, G., Low, K.L., and Bandyopadhyay, D.(2001) “Shader lamps: Animating real objects with image-based illumination”, In Proceedings of the 12th Eurographics Workshop on Rendering techniques, pp89-102.