Making any planar surface into a touch-sensitive display by a mere projector and camera

Jingwen Dai and Ronald Chung
Department of Mechanical and Automation Engineering
The Chinese University of Hong Kong, Shatin, NT, Hong Kong
{jwdai,rchung}@cuhk.edu.hk

Abstract

We address how an HCI (Human-Computer Interface) with small device size, large display, and touch input facility can be made possible by a mere projector and camera. The realization is through the use of a properly embedded structured light sensing scheme that enables a regular light-colored table surface to serve the dual roles of both a projection screen and a touch-sensitive display surface. A random binary pattern is employed to code structured light in pixel accuracy, which is embedded into the regular projection display in a way that the user perceives only regular display but not the structured pattern hidden in the display. With the projection display on the table surface being imaged by a camera, the observed image data, plus the known projection content, can work together to probe the 3D world immediately above the table surface, like deciding if there is a finger present and if the finger touches the table surface, and if so at what position of the table surface the finger tip makes the contact. All the decisions hinge upon a careful calibration of the projector-camera-table surface system, intelligent segmentation of the hand in the image data, and exploitation of the homography mapping existing between the projector's display panel and the camera's image plane. Extensive experimentation including evaluation of the display quality, touch detection accuracy, and system efficiency are shown to illustrate the feasibility of the proposed realization.

1. Introduction

HCI (Human-Computer Interface) has been traversing from firstly punch card and LEDs, then paper tape and CRO display, more recently mouse-plus-keyboard and LCD panel, and now fingers and touch-sensitive display panel over the history of development. Technologies have been ever improving, with the data-input mechanism growing only more natural, and the display only more vivid. Indeed for the input-output interface of computers, scarcely anything could be more natural than using our fingers to drag items on the "virtual desktop" of the computer, to open (and move and copy) files and folders, and to scroll (and enlarge) pages.

In today's computers and other portable devices like cellular phones and tablets, a large display panel is desired not only for enhancing display quality and coping with say aged vision, it is also essential, for touch input interface, for allowing finger - a rather bulky pointing device - to specify position on the "virtual" desktop in adequate precision. On that there is the following dilemma. A bigger and higher-resolution display, and a bigger keyboard, are desired to incur less strain on eyes and fingers. Yet they also make the devices less portable. This article attempts to solve this dilemma by exploring the possibility of replacing the display panel and the mouse-and-keyboard by a mere projector and camera. Specifically, it is to enable a light-colored table surface, to which the projection is illuminated, to serve as a touch-sensitive display panel for finger-based user input. The use of a projector in place of an LCD panel would dissociate display size from device size, making portability much less an issue. Touch-sensitive input facility on such a large display would also alleviate the need of a large keyboard.

The challenge is, from a single image alone there is generally difficulty in even distinguishing whether there is a physical contact between the finger and the table surface. The facility of acquiring certain 3D information about the illuminated workspace would be of much aid. A desirable way of making that possible is to use no additional sensor or instrument beyond what are already there - the projector and camera - by embedding structured codes into the projection. This way, the projector serves two purposes: the display device, as well as the 3D acquisition channel.

This paper aims at building the stated system, letting a regular tabletop surface to which the projection is illuminated become a touch-sensitive screen, with the entire system comprising a mere video projector and camera.

978-1-4673-1612-5/12/$31.00 ©2012 IEEE

1.1. Related Work

Barehand interface enjoys higher flexibility and more natural interaction than tangible interfaces. Earlier researches on barehand interfaces demanded assistance from some additional sensors [17, 9, 25, 5] installed on the surface.

With the development of computer vision algorithms, some vision-based projected tabletop interfaces equipped with finger tracking began to emerge in the last few years. Letessier [14] employed a single camera to detect and track the 2D position of the tip of bare finger on a planar display surface, but neglected finger clicking detection. In [7, 23], the "click" event was determined through a delay-based scheme, which has limited usability in applications that require fast response and multiple same-button clicks. Moreover, such click events were not intuitive and were rather deliberate since the user had to hold his finger over the button for a stipulated period to register a button select. Marshall [15] detected touch from the change in color of the fingernail when the finger was pressed against a surface. Song [19] proposed a finger-based interface in a projector-camera setting by examining if the finger and its shadow in the image were separated or merged. Wilson's PlayAnywhere [1] adopted extra infrared illumination to enhance the contrast between the finger and non-finger regions of the image data. This scheme however demands a capability of distinguishing the finger from its shadow robustly in the image. There is also substantial challenge in extending the scheme to multi-touch interface. Fitriani [10] projected a button based interface onto the surface of a soft deformable object such as a sofa pillow. The appearance changes of the virtual button being pressed were observed by a camera, which was considered as a signal for the touch event. The error detection rate was high due to complex and unpredictable deformations of the deformable surface.

After the release of PrimeSense's [20] depth-sensing camera-based Microsoft Kinect [18], depth-sensing cameras have been used in various interactive surface applications. LightSpace [3] used an array of depth-sensing cameras to track users's manipulations on multiple surfaces. In [2], the touch event was determined by using a per-pixel depth threshold derived from a histogram of the static scene. Omnitouch [6] detected surface touch by counting the pixel number in a flood filling operation in depth map. Yet depth-sensing camera is rather bulky, and is not a standard device as compared to pico-projector and CCD camera. All these hinders its applicability in hand-held consumer electronic products.

1.2. Main Contributions

This work aims at making the following contributions in building a touch-sensitive device: *(1) Using only off-the-shelf devices*: Pocket DCs and cellular phones with built-in projector and camera have already emerged in the consumable market. They form the necessary pro-cam foundation in building touch-sensitive interface in handheld devices. *(2) Achieving 3D sensing without explicit 3D reconstruction*: Detecting if a finger has indeed touched a tabletop surface and deciding at which position of the surface the touch takes place is a 3D sensing problem. Yet our system achieve all these without the need of going through explicit 3D reconstruction. The system exploits merely the homography mapping (induced by the table surface) between the projector's display panel and the camera's image plane. Without going through explicit depth recovery, the complexity of the sensing task is much reduced. *(3) Use of prior knowledge to enhance robustness*: By exploiting prior knowledge say about the relative geometry of the projector, camera, and projection surface, the system is endowed with better adaptability to environmental variations.

The remainder of this paper is structured as follows. In the next section, prior knowledge embraced in the pro-cam system is reviewed. In Section 3, the principle and strategy of embedding structured light codes in an invisible way into regular projection is described. The essential processes of the proposed method including hand segmentation, fingertip detection, and touch detection are detailed in Section 4. In Section 5, the system setup and experimental results are shown. Conclusion and possible future work are offered in Section 6.

2. Priors in Pro-Cam System

Consider a Pro-Cam system that has a projector illuminating certain display pattern to a planar projection surface (e.g. a tabletop surface) that is imaged by a camera. Once the two electronic instruments' intrinsic parameters and extrinsic relationship relative to the projection surface are fixed, the image data about the projection surface are predictable from the projection content. Specifically, which image position carries which part of the projection content that is reflected by the projection surface is governed by a particular homography mapping [22] existing between the projector's display panel Π_P and the camera's image plane Π_C, which is induced by the projection surface Π_T; and how close color or gray level in the image resembles that of the original projection content is governed by a radiometric process that can be calibrated. In this work, we make use of such priors for enhancing the efficiency and precision of the human-computer interface we aim at building.

2.1. Homography Estimation

There are altogether three homographies in our system: the homography H_{TC} between the camera's image plane Π_C and table surface Π_T, the homography H_{PT} between the projector's display panel Π_P and Π_T, and the homography H_{CP} between Π_C and Π_P that is induced by the table

surface. Among them, H_{PT} is used for projector keystone correction, H_{CP} is for retrieval of the structured light code, and H_{CT} is for deriving H_{PT} which cannot be directly calibrated for the reason that projector does not have visual sensing capability.

Since homography can be expressed as a 3×3 matrix of arbitrary scale, i.e., a matrix with 8 degrees of freedoms (DOFs), it could be determined from as few as four pixel correspondences only across the input and output planes; when more than four correspondences are available, least-squares solution of the homography is to be obtained.

Firstly, the homography H_{TC} between the camera's image plane and the table surface is determined. On this, any rectangular object of known or standard dimension (e.g. credit card, plastic ruler) placed on the projection surface can be used as the calibration object. The H_{TC} could be estimated as

$$X_T = H_{TC} X_C, \tag{1}$$

where X_T is any corner of the flat reference object in homogenous coordinates, and X_C is the corresponding point on the camera's image plane.

With H_{CT}, the homography H_{CP} between the camera and projector could be derived with ease. By instructing the projector to project some distinct markers (e.g. chessboard) to the table surface, the homography could be calculated in the same way as the above:

$$X_C = H_{CP} X_P, \tag{2}$$

where X_C is the position of projected marker in the observed image, and X_P is the marker position on the display panel of the projector, both in homogeneous coordinates.

Finally, the homography H_{TP} between projector and table surface is determined as

$$X_T = H_{TC} X_C = H_{TC} H_{CP} X_P = H_{TP} X_P. \tag{3}$$

2.2. Radiometric Prediction

Besides the geometric distortion, the photometric appearance (e.g. brightness, RGB color etc.) of the projection surface in the image data is another prior that has to be seized before it can be exploited in the construction of the touch interface system. The appearance is generally distorted from that of the projected pattern due to nonlinearity of the projection and imaging processes, the texture of the projection surface, and the influence of ambient illumination. To predict the projection appearance in the image data, radiometric calibration is necessary.

Here we employed the photometric model described in [26], which is formulated as

$$\mathbf{C}_{pre} = \mathbf{V}\mathbf{P} + \mathbf{F}, \tag{4}$$

where $\mathbf{C}_{pre}$ and $\mathbf{P}$ are the RGB values of the predicted image and real image respectively, the 3×3 matrix $\mathbf{V}$ is the color mixing matrix that captures all the couplings between the projector and camera channels and their interaction with the spectral reflectance of the projection surface, and the vector $\mathbf{F}$ is the contribution of the environmental lighting relative to the black level of the projector.

To measure these parameters, five images are projected and captured, first a black image, then a red, a green, a blue and a chromatic in sequence. In addition to the one image projection and two image captures required in homography estimation, the process of deriving all the priors involves 7 projection-capture cycles, which can be accomplished in only a few seconds. Unless the system is moved to another working environment, or the environmental illumination is changed, the prior knowledge is approximately constant in the operation of the touch interface system.

3. Embedding Codes into Video Projection

3.1. Imperceptible Structured Light

The fundamental principle behind imperceptible structured code embedding [24, 4] is the temporal integration process achieved by projecting each image twice at high frequency: a first image I containing the actual code information (e.g. by adding or subtracting a certain amount (Δ) to or from the pixels of the original image depending on the code), and a second image I' that compensates the distortion in the first image with the goal that the two quick projections as a whole would deliver an overall visual perception that is without the embedded code. More precisely, if images I and I' are shown to human subject at a rate double that of the fastest rate (the flicker fusion threshold) human vision can differentiate temporally, the collective human visual perception would be merely the average of I and I'.

In the case of color projection, it is possible to embed n-nary structured light code (where $n > 2$) into the three different channels (R,G,B). However, in this work, for simplicity and for enhancing the robustness to noise, we use $n = 2$, i.e., we use only binary code and embed it into all three color channels simultaneously. Let B, O, I and I' be the binary code to be inserted, the original image, the first projected image, and the second complementary image respectively. Then the projected image and the complementary image could be expressed as

$$I_i(x,y) = O_i(x,y) + P(x,y), \tag{5}$$

$$I'_i(x,y) = O_i(x,y) - P(x,y), \tag{6}$$

$$P(x,y) = \begin{cases} \Delta, & when \quad B(x,y) = 1; \\ 0, & when \quad B(x,y) = 0. \end{cases} \tag{7}$$

where $i = \{\text{R,G,B}\}$ indicates whether it is the red, green, or blue channel, and Δ is the embedded intensity corresponding to bit 1 in the structured light code.

In order to achieve imperceptible structured light projection, the frequency of projection must exceed the flicker fusion threshold, which is $75Hz$ for most of the people. Here we take one projection-capture cycle as an example to explain the strategy of projector-camera synchronization. Firstly, we ensure that the projector projects an image every $10ms$, i.e., at $100Hz$. Along the time axis, the projected image I and the complementary image I' are projected at the time instants $0ms$, $10ms$ respectively. With a refresh rate of the camera at about 100 frames per second, the camera captures the images C and C' at $5ms$ and $15ms$ shortly after the projector projects the projected image and complementary image to the scene. At $20ms$ a new projection-capture cycle will begin. With the aforementioned projection-capture strategy, the system could capture 50 image pairs per second.

Notice that the embedded codes could be internally and simply extracted from the "subtraction image" between consecutively captured images, as

$$S(x,y) = \max_i [C_i(x,y) - C'_i(x,y)], i = \{R,G,B\}. \quad (8)$$

3.2. Embedded Pattern Design Strategy and Statistical Analysis

Structured light coding is about equipping each pattern position a unique code that can be distinguished in the image data. The coding can be realized over time or space (the 2D space of the code pattern). In the touch sensitive interface we are to build, the movement of hand and finger, the real-time operation requirement, and the constraints of imperceptible code embedding make the temporal coding scheme not applicable. We are thus left with the option of using spatial coding scheme, which has the advantage that 3D determination can be achieved with at few as one single image.

Since the resolution, optical parameters, and the position and orientation with respect to the target object are all different between the camera and projector, it is impossible to align the pixels on the camera's image plane and those on the projector's display panel for one-to-one pixel correspondence. To overcome the problem, binary spatial coding methods generally adopt some special shapes (such as stripe, square, circle etc.) as appearance profiles, which could be easily segmented in the decoding stage. A shortcoming of this design scheme is that the density of the effective feature points is sparse, and in our case is generally too sparse to ensure that the depth information of the fingertip can always be derived no matter where it is located. Some typical methods in binary spatial coding are listed in Table 1. In the literature there is not an effective method to generate a binary array of 640×480 size that has the required unique window property. For this reason, in this work we seek to generate the pattern array by statistical analysis.

Method	Array Size	Win. Size	Alph. Length
Salvi [16]	29×29	3×3	3
Spoelder [12]	65×63	2×3	2
Albitar [13]	27×29	3×3	3
Desjardins [8]	53×38	3×3	3

Table 1. Summary of typical spatial coding methods

In our system, we use a pico projector that is of 640×480 resolution. To make sure that every pixel has a unique binary code, it is required that $2^k \geq 640 \times 480$, which means $k \geq \ln(640 \times 480) \geq \ln 2 \geq 18.23$. In other words, the codeword at each pattern position must be at least 18 bits long. In accordance with the resolution of pico projector, a matrix of 640×480 is to be filled with pseudo-random generated sequence consisting of 0 and 1 in standard uniform distribution. If an $m \times n$ window is selected for coding each pixel, and if the window is picked to be the one with the pixel as its bottom-right corner, totally $(640 - m + 1) \times (480 - n + 1)$ pixels will be coded by a (mn)-bit binary string. The codeword of every effective pixel can be derived and some statistical analysis can be employed to evaluate the code uniqueness. For our pico projector, random generation of 6×6 arrays are generally sufficient to equip each pixel with a unique window label.

In our experimentation, after conducting 100 trials of pattern generation, the array with the largest average inter-codeword Hamming distance ($\bar{H} = 4.524$) was derived. The large inter-codeword Hamming distance corresponds to good noise-tolerance of the codewords on the imaging side. We chose this array to embed into the normal video projection.

In the decoding stage, the correspondences between the camera's image plane and the projector's display panel were established by the homography induced by the projection surface. This will be discussed in the following section in depth.

4. Touch Detection using Homography and Embedded Code

For the purpose of locating the position of the fingertip and determining whether a physical touch takes place, some preliminary processes need be employed, such as hand segmentation and fingertip detection. In this section we discuss these processes in the circumstance of our particular pro-cam system.

4.1. Hand Segmentation and Fingertip Detection

The aforementioned priors about the pro-cam system allows us to know where the video contents are projected and how they should appear in the image data. With the priors, simple background subtraction is generally adequate to

segment the hand as the set of pixels that are out of expectation on the projection surface. However, hand could come with shadow which might partially come out from the background subtraction process. We adopt a coarse-to-fine scheme to eliminate the shadow's influence.

After subtracting the geometrically and photometrically predicted background image from the actual image captured by the camera, an approximate region of the hand can be extracted as the collection of pixel positions where the difference magnitudes exceed a threshold. An example is shown as the yellow curve in Fig. 1a. In this region, except the shadow subregion of it, the influence caused by video projection has been minimized. The actual hand portion and the shadow portion of the region can be distinguished from the fact that the hand portion as flesh generally appears much brighter than the shadow portion. As a consequence, the hand subregion is generally more salient in the H-channel (Fig. 1b) of the HSV color space. With the aid of this constraint, the hand subregion can be refined by local adaptive threshold segmentation [21]. After filling up the isolated small cavities by the use of the morphological "close" operation, the largest connected subregion is regarded as the hand (Fig. 1c).

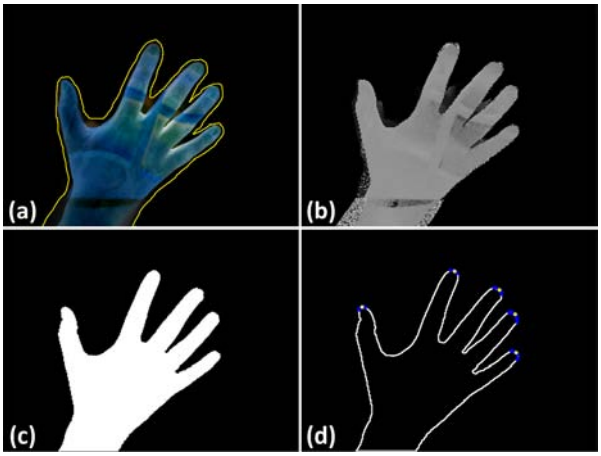

Figure 1. (a) Approximate segmentation in subtraction image, (b) H-channel, (c) refined hand region, (d) hand contour and detected fingertips

The extracted contour of the hand (Fig. 1 serves to offer fingertip candidates through a simple arc line analysis. Let $T(x), x = 1, \ldots, N$ be the various points of the hand silhouette in clockwise order, where N is the total number of contour points. Whether a particular contour point $T(k)$ is a fingertip candidate is examined by the curvature of the contour there. We express the curvature approximately as the angle θ between the vectors $v_1 = T(k) - T(k-t)$ and $v_2 = T(k) - T(k+t)$, where $T(k-t)$ and $T(k+t)$ are contour points in the vicinity of $T(k)$, each on a different side of $T(k)$ at an interval of t points from it. If $\theta < \frac{\pi}{2}$ and $|v_1, v_2| > 0$, $T(k)$ is regarded as a fingertip candidate. The second conditional term as a determinant is employed to distinguish fingertip peaks from valleys between two fingers. Finally, candidates that are consecutive or nearly consecutive in the hand silhouette are clustered into the same group (blue points in Fig. 1d), and in each group only the candidate in the median position is confirmed as a finger tip (yellow points in Fig. 1d).

4.2. Touch Detection Through Homography

With the fingertips detected, the next task is to examine if any of the fingertips touches the display surface. In the coding design, we ensure that every pixel in the projected pattern is coded by a 36-bit binary codeword. However, as discussed above, it is impracticable to align the pixels on the camera's image plane and those on the projector's display panel for one-to-one pixel correspondence between the two. Instead, we make use of the homography between the image plane and display panel that is induced by the table surface. Below we use the single-touch case as an example to illustrate how a mere touch is detected. Multi-touch is a simple extension of the single-touch.

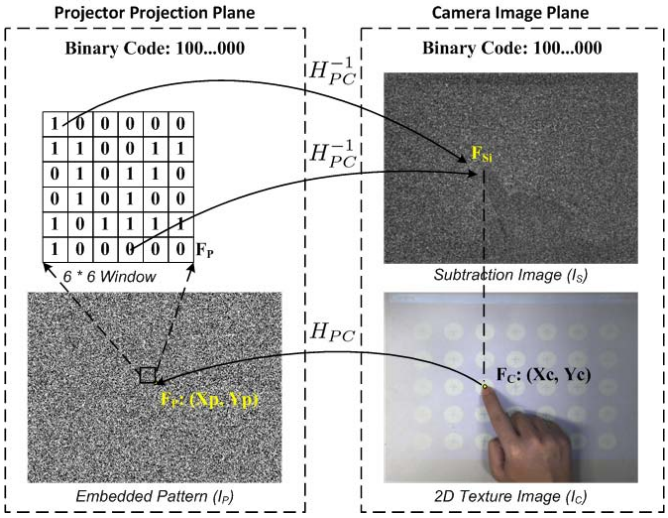

Figure 2. Touch detection via homography

As illustrated in Fig. 2, suppose we have a finger touching the projection surface. The fingertip F_C lies on the plane of the projection surface, and thus would satisfy the associated homography. More precisely, a position F_P on the display panel of the projector Π_P can be derived in homogenous coordinates as $\tilde{F_P} = H_{PC}\tilde{F_C}$. The codeword at F_P is then determined by the code values of the pixels F_{P_i} in a 6×6 window that has F_P as its bottom-right corner. In other words, the binary codeword BC_P at F_P is regarded as

$$BC_P = \sum_{i=0}^{35} 2^i \cdot I_P(F_{P_i}), \tag{9}$$

where $F_{P_i} \in \{(X_{P_i}, Y_{P_i}) | X_P - 5 \leq X_{P_i} \leq X_P, Y_P - 5 \leq Y_{P_i} \leq Y_P\}$.

On the other hand, the binary code embedded in the image data at point F_C can be observed as

$$BC_S = \sum_{i=0}^{35} 2^i \cdot I_S(F_{S_i}), \quad (10)$$

$$\tilde{F_{S_i}} = H_{PC}^{-1} \tilde{F_{P_i}}, \quad (11)$$

where $\tilde{F_{S_i}}$ and $\tilde{F_{P_i}}$ are homogenous representations.

If the Hamming distance between BC_P and BC_S is less than a preset threshold λ_H, F_P and F_S are considered as sharing the same code, meaning that the touch has taken place. Otherwise, the finger is regarded as not having physical contact with the table surface. The threshold λ_H should be adjusted according to the ambient illuminations for suitable noise-tolerance.

The above allows touch to be determined without going through explicit 3D reconstruction, and can operate in real-time.

4.3. From Resistive Touch to Capacitive Touch or Floating Touch

In the last section, we have emulated a "resistive" touch operation, which requires touch action with a certain pressure on the projection surface. Below we show how to enhance the touch sensitivity and move the interface from a "resistive touch" to a "capacitive touch" or even a slight "floating touch".

In fact we can generate from the table surface-induced homography to another homography that is induced by a plane parallel to but slightly elevated from the table surface, as indicated by any of the shown dashed lines in Fig. 3). The dash lines correspond to different levels of touch sensitivity demanded. If the homography so generated is satisfied by any detected finger tip in the image data, a touch action can be regarded as confirmed.

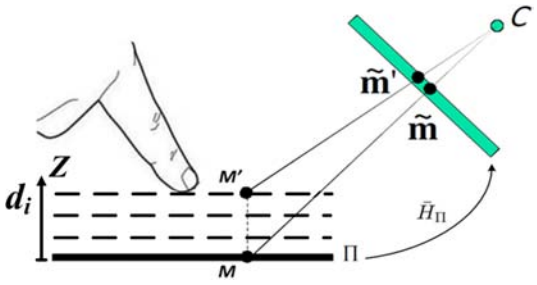

Figure 3. Homography transfer across parallel planes.

As shown in Fig. 3, given a plane Π, we can define a coordinate frame $W : X - Y - Z$ local to it, with X, Y axes within the plane Π and Z-axis perpendicular to Π. Suppose the plane Π is the real table surface, and we know the homography $\bar{H}_\Pi$ from Π to the camera's image plane that is induced by Π itself. Then let the pre-calibrated projection matrix of the camera be

$$P \cong [p_1, p_2, p_3, p_4] \cong K[r_{1\Pi}, r_{2\Pi}, r_{3\Pi}, t_\Pi], \quad (12)$$

where K is the 3×3 matrix containing all the intrinsic parameters of the camera. Notice that the homography $\bar{H}_\Pi$ that owns the property:

$$\tilde{m} \cong \bar{H}_\Pi [X, Y, 1]^T, \quad (13)$$

is related to the camera projection matrix by $\bar{H}_\Pi \cong [p_1, p_2, p_4] \cong K[r_{1\Pi}, r_{2\Pi}, t_\Pi]$.

Consider a plane Π_{d_i} that is parallel to but elevated from Π by a perpendicular distance d_i. For the 3D position (X, Y, d_i) on Π_{d_i}, which is elevated from point $(X, Y, 0)$ on Π perpendicularly by distance d_i, the image projection $\tilde{m}'$ can be expressed as

$$\begin{aligned} \tilde{m}' &\cong K[R_\Pi, t_\Pi][X, Y, d_i, 1]^T \\ &\cong K(X r_{1\Pi} + Y r_{2\Pi} + d_i r_{3\Pi} + t_\Pi) \\ &\cong K([r_{1\Pi}, r_{2\Pi}, t_\Pi] + d_i[0, 0, r_{3\Pi}])[X, Y, 1]^T \\ &\cong (\bar{H}_\Pi + d_i[0, 0, K r_{3\Pi}])[X, Y, 1]^T, \end{aligned} \quad (14)$$

By substituting Eq. 13 into Eq. 14, we have

$$\tilde{m}' \cong (I + d_i[0, 0, p_3]\bar{H}_\Pi^{-1})\tilde{m} \cong H_{Cd_i}\tilde{m}. \quad (15)$$

Hence, through the original homography and the third column of the camera projection matrix, we can derive the homography H_{Cd_i} between the camera's image plane and the elevated plane. In a similar way, the homography H_{Pd_i} between projector's display panel and the elevated plane can also be expressed. Finally, the new homography between the projector's display panel and the camera's image plane, that is induced by elevated plane, is obtained as $H_{CPd_i} = H_{Cd_i} H_{CP} H_{Pd_i}^{-1}$, which can be adopted for more sensitive touch sensing on the table surface.

5. Experiments

In order to assess the feasibility of the described system for barehand human-computer interface, we conducted experiments to evaluate display quality, touch detection accuracy, and system efficiency respectively.

The projector-camera system we used in our experiment consisted of a DLP projector with a native resolution of 640×480 and an interface for firmware configuration (TI DLP Pico Projector Development Kit 2), plus a camera of 648×488 resolution at 120fps (Point Grey FL3-FW-03S1C camera with Myutron FV0622 f6mm lens), both being off-the-shelf equipments. The system was configured for a working distance of about $500mm$, making a 15-inch projection area. If short-throw projector and short focus lens are employed, a bigger projection area could be acquired with shorter distance.

5.1. Display Quality Evaluation

Embedded code imperceptibility and user satisfaction is of the first priority in the system design. We conducted user studies based on a questionnaire. Twenty persons were invited to participate in this experiment. 500 images were collected from Google Image randomly, in which binary pattern was embedded with different intensities. The viewers were seated in front of a desk surface where the video contents were projected, and asked to comment on the quality of the image. The questions asked were simplified from the questionnaire in [11], focusing on the feeling of flickering, the recognition of image deterioration, and the overall satisfaction for projection quality. The score for each question ranged from 0 to 10.

The average scores of the subjective evaluation are illustrated in Fig. 4. When the embedded intensity was small, i.e., $\Delta = 5, 10$, the viewer could rarely notice the embedded codes and were satisfied with the projection quality. With the increase of the embedded intensity, the viewers' sense of flickering and image degradation became stronger. When $\Delta = 25$, almost every viewer was not satisfied with the projection quality.

In practice, because it was difficult to retrieve weakly embedded codes with the standard commercial cameras, we chose $\Delta = 10$ in our configuration, striking a compromise between user satisfaction and code imperceptibility.

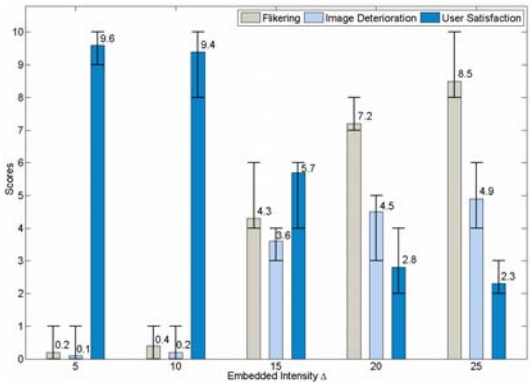

Figure 4. User studies results for code imperceptibility

5.2. Touch Accuracy Evaluation

Similar to [6], we specially designed an image, in which 35 circles were distributed uniformly. As shown in Fig. 5a, the center of each circle, indicated by the cross symbol, was known. The testing pattern was projected to three table surfaces with different textures as shown in Fig. 5b-d. In each round, the users clicked the virtual projected circles one by one as accurately as they could. If a touch contact was detected, a yellow circle was placed around the clicked circle (Fig. 5b & d). Five persons were invited to participate in the experiment, each of them conducted 6 rounds (on the three surfaces and under two ambient illuminations). Totally, 1050 touch trials were produced.

Surface	Illumination			
	Dark		Normal	
	ϵ(px)	FRR/FAR(%)	ϵ(px)	FRR/FAR(%)
Gray	2.98	1.12/0.45	3.05	1.32/0.48
Yellow	3.04	1.23/0.57	3.12	1.54/0.61
Artifact	3.12	1.77/0.67	3.20	1.76/0.63

Table 2. The quantitative experiment results

The precision of touch position localization is evaluated by the average distance between ground-truth and the detected position, which is formulated as $\epsilon = \frac{1}{N_t}\sum_{i=1}^{N_t}\sqrt{(X_{d_i}-X_{g_i})^2+(Y_{d_i}-Y_{g_i})^2}$, where N_t is the total number of correctly detected touch contacts, and (X_{d_i}, Y_{d_i}) and (X_{g_i}, Y_{g_i}) are the detected position and ground-truth respectively.

The accuracy of touch detection is estimated by false reject rate (FRR), the probability that the system fails to detect an actual touch action, and false accept rate (FAR), the probability that the system incorrectly confirms a non-contact action as a touch contact. FRR and FAR are formulated as $FRR = N_{md}/N, FAR = N_{fd}/N$, where N is the total trial number, N_{md} and N_{fd} are the number of missed detections and false detections respectively.

The detailed quantitative testing results, listed in Table 2, illustrate the performance and robustness of the described system against different projection surfaces and different surrounding illuminations. Here, we compared our method with some recent depth-camera sensing based methods. In [2], the informal observed spatial error of finger detection on planar surface was between 3-6 pixels, but the finger click detection error was not mentioned. As for Omni-Touch [6], the FRR and FAR of finger click detection on four different surfaces were reported as 0.8% and 3.3%. Even though the evaluation data-sets, the sensing systems and working environments were not all exactly identical, the comparison results show that the described system has at least comparable performance even under less complicated devices. Due to page space constraints, more results on finger dragging tracking and multi-touch capability are shown in the supplementary video.

5.3. Efficiency Evaluation

For human-computer interface, real-time performance is of great importance. Hence we implemented the proposed system in C++ using the Intel OpenCV Library to evaluate its processing time. Through multi-thread programming, the projection-capture process and calculation process were executed in two different threads respectively, each of which was able to run in real time in a desktop computer with Intel Core2 Duo 2.53GHz CPU. Table 3 shows

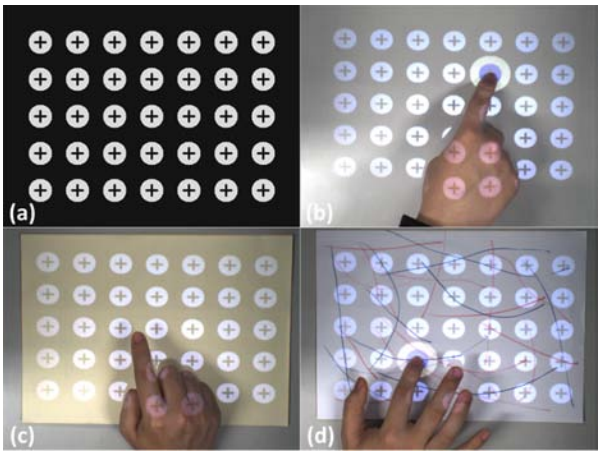

Figure 5. (a)Image projected for ground-truth collection, (b) gray surface, (c) yellow surface, (d) surface with artifacts

Subroutine	Hand Seg.	FTip Loc.	Touch Det.	Total
Time (ms)	14.63	1.32	1.74	17.69

Table 3. Average processing time

the average processing times for hand segmentation in 2D image, fingertip localization, and touch detection. The total time consumption is less than $20ms$, indicating the system meets the requirement of real-time application.

6. Conclusion and Future Work

This article explores the possibility of replacing the display panel and the mouse-and-keyboard by a mere projector and camera. Specifically, it is to enable a light-colored table surface, to which the projection is illuminated, to serve as a touch-sensitive display panel for finger-based user input.

The described work lays down the setup and design of the pro-cam system for touch-sensitive interface. Single-touch, touch dragging tracking and multi-touch facilities are also constructed and thoroughly experimented with. All these form the basis of a more complete touch interface system. Future work includes more thorough experimentation with multi-hand interface using the system. Based upon the touch detection facility, advanced touch gestures (e.g. double clicking, scrolling, zoom-in, zoom-out) and even typing recognition on the described platform will also be studied.

Acknowledgement

This work is affiliated with CUHK MoE-Microsoft Key Laboratory for Human-centric Computing & Interface Technologies.

References

[1] A. D. Wilson. Playanywhere: a compact interactive tabletop projection-vision system. In *ACM UIST*, 2005. 2

[2] A. D. Wilson. Using a depth camera as a touch sensor. In *ACM ITS*, 2010. 2, 7

[3] A. D. Wilson and H. Benko. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *ACM UIST*, 2010. 2

[4] A. Grundhoer, M. Seeger and et al. Dynamic adaptation of projected imperceptible codes. In *IEEE ISMAR*, 2007. 3

[5] C. Harrison, D. Tan and D. Morris. Skinput: Appropriating the body as an input surface. In *ACM CHI*, 2010. 2

[6] C. Harrison, H. Benko and A. D. Wilson. Omnitouch: wearable multitouch interaction everywhere. In *ACM UIST*, 2011. 2, 7

[7] C. von Hardenberg and F. François. Bare-hand human-computer interaction. In *ACM PUI*, 2001. 2

[8] D. Desjardins and P. Payeur. Dense stereo range sensing with marching pseudorandom patterns. In *CRV*, 2007. 4

[9] D. Leigh and P. Dietz. Diamondtouch characteristics and capabilities. In *UbiComp*, 2002. 2

[10] Fitriani and G. Wooi-Boon. Interacting with projected media on deformable surfaces. In *IEEE ICCV*, 2007. 2

[11] H. Park, B. Seo and J. Park. Subjective evaluation on visual perceptibility of embedding complementary patterns for nonintrusive projection-based augmented reality. *IEEE Trans. Circuits Syst. Video Techno*, 20(5):687–696, 2010. 7

[12] H. Spoelder, F. Vos and et al. Some aspects of pseudo random binary array-based surface characterization. *IEEE Trans. Instrum. Meas.*, 49(6):1331–1336, 2000. 4

[13] I. Albitar, P. Graebling and C. Doignon. Robust structured light coding for 3d reconstruction computer vision. In *IEEE ICCV*, 2007. 4

[14] J. Letessier and F. Berard. Visual tracking of bare fingers for interactive surfaces. In *ACM UIST*, 2004. 2

[15] J. Marshall, T. Pridmore and et al. Pressing the flesh: Sensing multiple touch and finger pressure on arbitrary surfaces. In *Pervasive Computing*, 2008. 2

[16] J. Salvi, J. Batlle and E. Mouaddib. A robust-coded pattern projection for dynamic 3d scene measurement. *Pattern Recognition Letters*, 19:1055–1065, 1998. 4

[17] Light Touch. http://lightblueoptics.com. 2

[18] Microsoft Kinect. http://www.xbox.com/kinect. 2

[19] P. Song, S. Winkler and et al. Vision-based projected tabletop interface for finger interactions. In *Proc. of CHI*, 2007. 2

[20] PrimeSence. http://www.primesense.com. 2

[21] R. Gonzalez and R. Woods. *Digital Image Processing(3e)*. Prentice Hall, 2008. 5

[22] R. Hartley, and A. Zisserman. *Multiple View Geometry in Computer Vision(2e)*. Cambridge University Press, 2004. 2

[23] R. Kjeldsen, C. Pinhanez and et al. Interacting with steerable projected displays. In *IEEE AFGR*, 2002. 2

[24] R. Raskar and et al. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *ACM SIGGRAPH*, 1998. 3

[25] J. Rekimoto. Smartskin: An infrastructure for freehand manipulation on interactive surfaces. In *ACM CHI*, 2002. 2

[26] X. Chen, X. Yang and et al. Color mixing property of a projector-camera system. In *PROCAMS*, 2008. 3