Touchscreen Everywhere: On Transferring a Normal Planar Surface to a Touch-Sensitive Display

Jingwen Dai, *Member, IEEE,* and Chi-Kit Ronald Chung, *Senior Member, IEEE*

***Abstract*—We address how a 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 3-D workspace 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 on the table surface the contact is made. 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, hand segmentation accuracy, touch detection accuracy, trajectory tracking accuracy, multitouch capability and system efficiency are shown to illustrate the feasibility of the proposed realization.**

***Index Terms*—Accuracy evaluation, hand segmentation, homography, imperceptible structured light embedding, touch detection, touch-sensitive display.**

I. Introduction

HUMAN-COMPUTER interface (HCI) 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 PDAs, 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, on 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 3-D 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 3-D acquisition channel.

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

II. Related Work

Traditional HCI is largely mouse and keyboard based, which is effective but not necessarily the most natural. Tangible interfaces have been used in some projected environments. By letting users hold some physical objects in hand and manipulate them, more comfortability could be induced in

Manuscript received November 6, 2012; revised April 18, 2013 and July 20, 2013; accepted September 18, 2013. Date of publication November 1, 2013; date of current version July 15, 2014. This paper is an extended version of the authors' PROCAMS2012 paper [1]. This paper was recommended by Associate Editor B. W. Schuller.

J. Dai is with the Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 USA (e-mail: dai@cs.unc.edu).

C.-K.R. Chung is with the Vocational Training Council of Hong Kong, Wan Chai, Hong Kong (e-mail: rchung@vtc.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2013.2284512

the interaction. Sensetable [2] uses a projected interface for visualization and design. Physical objects with embedded sensors can be held by users for movements to represent the corresponding interactions. The Flatland system [3] projects onto a whiteboard, and interactions are based on the interpretation of strokes via the stylus onto the whiteboard. Escritoire [4] uses special pens with embedded sensors to enable interaction between user and an illuminated table surface. More recently, Jones [5] demonstrates a projector-camera system that acquires the object geometry and enables direct interaction through an IR tracked stylus. These applications are all based on manipulating tangible objects like pens for interactions. The flexibility can however be further improved if even the intermediate objects can be waived, and hands and fingers are directly used. To many, barehand interface enjoys higher flexibility and more natural interaction than tangible interfaces.

Earlier researches on barehand interfaces demanded assistance from some additional sensors. The interfaces in DiamondTouch [6] and SmartSkin [7], both allow hand input on a table surface, but the table has to embed a grid of wired sensors in the first place. SmartSkin recognized multiple hand positions and shapes and calculated the distance between the hand and the surface by using capacitive sensing and a mesh-shaped antenna. Light Touch [8] employed an infrared sensor to recognize finger's contact with the projection surface. Bonfire [9] detected finger tapping using the laptop's on-board accelerometer. Skininput [10] resolved the location of finger taps on the arm and hand by analyzing mechanical vibrations that propagate through the body. These signals were collected by a array of sensors worn as an armband.

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. Katz [11] proposed a framework for a multitouch surface using multiple cameras. The touch detection problem was addressed by installing a second camera of which the optical axis is parallel to the projection surface. The additional instrument as the second camera will increase system complexity and the configuration time when the user want to move the system to other places. Letessier [12] employed a single camera to detect and track the 2-D position of the tip of bare finger on a planar display surface, but neglected finger clicking detection. In [13], [14], 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 [16] proposed a finger-based interface in a projector-camera setting that examines if the finger and its shadow in the image are separated or merged. Wilson's PlayAnywhere [17] 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 multitouch interface. Fitriani [18] 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 of the touch event. The error detection rate was high due to complex and unpredictable deformations of the deformable surface.

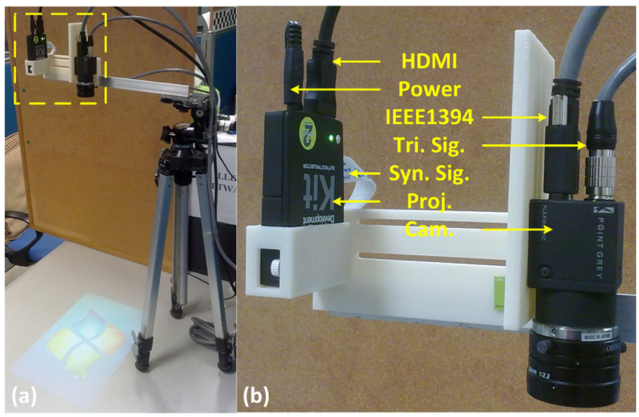


Fig. 1. System prototype.

After the release of PrimeSense's depth-sensing camera-based Microsoft Kinect, depth-sensing cameras have been used in various interactive surface applications. LightSpace [19] used an array of depth-sensing cameras to track user's manipulations on multiple surfaces. In [20], the touch event was determined by using a per-pixel depth threshold derived from a histogram of the static scene. Omnitouch [21] detected surface touch by counting the pixel number in a flood filling operation in the depth map. Although the PrimeSense's next-generation 3-D sensor Capri, released in 2013, can fit into smaller devices, but it is still not a standard device as compared to pico-projector and CCD camera. All these hinder its applicability in hand-held consumer electronic products.

III. System Overview

The prototype of proposed system is illustrated in Fig. 1. The projector-camera system consists 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 120 frames/s (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 $500mm$, targeting at 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.

The projector and a camera were mounted rigidly and then were fixed on a tripod standing on a table surface, as shown in Fig. 1(a). The projector and camera were connected to a desktop computer through HDMI and IEEE1394 interfaces respectively, and the hardware trigger signal of the camera was connected to the sync. output of the projector for synchronization between them, which are illustrated in Fig. 1(b). Moreover, the projector-camera system was precalibrated using the method detailed in [22].

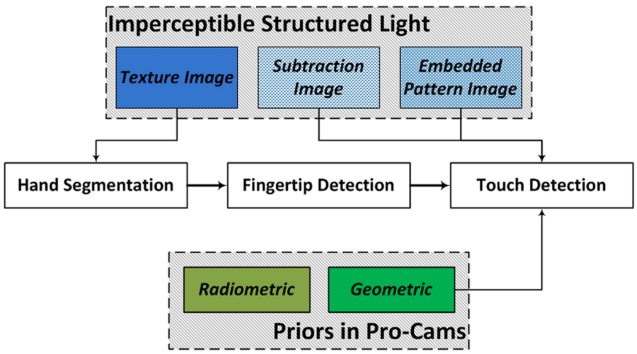


Fig. 2. System flowchart.

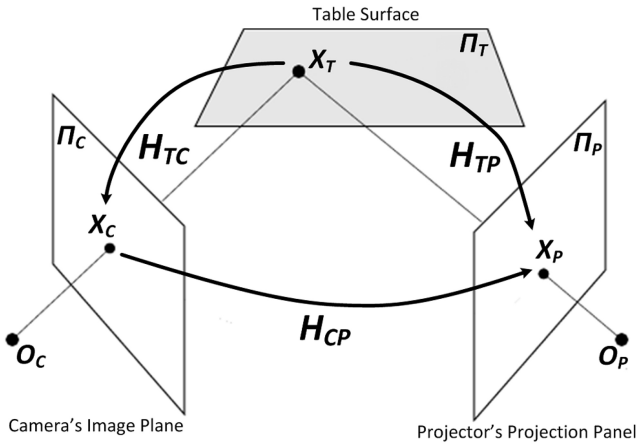


Fig. 3. Homographies in projector-camera-surface system.

The system flowchart is shown in Fig. 2. Under the framework of imperceptible structured light sensing, the texture image and subtraction image containing embedded codes are acquired, and the embedded pattern image is known by pattern design. Besides, some prior knowledge is embraced in the projector-camera system, such as the geometric relationship between the projector, camera and table surface, and the radiometric properties of the instruments, table surface and ambient light. The proposed system makes use of the image data provided by imperceptible sensing and priors embraced in the projector-camera system to detect the touch action. Firstly, the hand region is segmented from the texture image. Then the positions of fingertip in 2-D image are localized in the binarized hand region. Whether the finger is touching the surface is determined by comparing the binary codes derived from the subtraction image and embedded pattern image through the associated homography mapping.

This paper 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 projector-camera foundation in building touch-sensitive interface in handheld devices.
2) *Achieving 3-D sensing without explicit 3-D 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 3-D sensing problem. Yet our system achieve all these without the need of going through explicit 3-D 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) *Precise hand segmentation in projector-camera system*
   Combining contrast saliency and region discontinuity, the coarse-to-fine approach can achieve robust, precise and also rapid hand-segmentation, without the need of pretraining and precalibration procedures.
4) *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 projector-camera system is reviewed. In Section V, 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 VI. In Section VII, the system setup and experimental results are shown. Conclusion and possible future work are offered in Section VIII.

IV. Priors in Projector-Camera System

Consider a projector-camera 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 [23] 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. In this paper, we make use of such priors for enhancing the efficiency and precision of the human-computer interface we aim at building.

As shown in Fig. 3, 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 retrieving 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 eight degrees of freedoms (DOFs), it could be determined from as few as four pixel correspondences across the input and output planes; when more than four correspondences are available, the least-squares solution of the homography is to be used.

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 the projector and table surface is determined as

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

V. Embedding Codes into Video Projection

A. Imperceptible Structured Light

The fundamental principle behind imperceptible structured code embedding [24]–[26] 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 upon the polarity of the code if binary coding is used], 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 paper, 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

$$I_i(x, y) = O_i(x, y) + P(x, y) \tag{4}$$

$$I'_i(x, y) = O_i(x, y) - P(x, y) \tag{5}$$

$$P(x, y) = \begin{cases} \Delta, & when \quad B(x, y) = 1; \\ 0, & when \quad B(x, y) = 0 \end{cases} \tag{6}$$

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.

Notice that the embedded codes could be internally and simply extracted from the subtraction image [1] between consecutively captured images, as

$$S(x, y) = \max_i [C_i(x, y) - C'_i(x, y)], \quad i = \{R, G, B\}. \tag{7}$$

For the detailed projector-camera synchronization strategy, please refer to [26].

B. Embedded Pattern Design Strategy and Statistical Analysis

Structured light coding is about equipping each pattern position with a unique code that can be distinguished in the image data. The coding can be realized over time or space (the 2-D space of the code pattern itself). In the touch sensitive interface we aim at building, the fast 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 the spatial coding scheme, which has the advantage that 3-D determination can be achieved with as few as a single image.

Since the resolution, optical parameters, and the position and orientation of the camera and projector with respect to the target object are generally different, it is difficult to align pixels on the camera's image plane to those on the projector's display panel for exact one-to-one pixel correspondence. To overcome the problem, binary spatial coding methods generally adopt certain specific shape primitives (such as stripes, squares, circles etc.) as appearance profiles, which are readily to be segmented from the image data 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. Here we propose a new binary encoding scheme that allows to achieve pixel-level precision.

Almost all the spatial coding methods were based on perfect map or M-array theory for their unique window property. MacWilliams [27] and Etzion [28] proposed methods to construct M-array mathematically. By folding pseudorandom array, the methods are effective and efficient to generate M-arrays. However, they could only generate the ones of $n_1 \times n_2$ size with the $k_1 \times k_2$ window property, where $n = 2^{k_1 k_2} - 1$,

[1] All the subtraction images in this paper are scaled to [0, 255] for illustration purposes.

TABLE I
SUMMARY OF TYPICAL SPATIAL CODING METHODS

Method	Array Size	Win. Size	Alph. Length
Morita [30]	24×24	3×4	2
Kiyasu [31]	18×18	4×2	2
Salvi [32]	29×29	3×3	3
Spoelder [33]	65×63	2×3	2
Albitar [34]	27×29	3×3	3
Desjardins [35]	53×38	3×3	3
Chen [36]	82×82	3×3	7

$n_1 = 2^{k1} - 1$, $n_2 = n/n_1$. In our case, the resolution of pico projector is 640×480. To ensure that every pixel has a unique binary code, $2^k \geq 640 \times 480$, meaning that $k \geq \ln(640 \times 480)/\ln 2 \geq 18.23$. Thus the windows size should be set as 5×5. As a result, the dimension of the perfect map is $n_1 = 2^{k_1} - 1 = 31$, $n_2 = n/n_1 = 1082401$. However, this result is not applicable to our projector.

Some researchers employed other practical methods to generate the perfect map. Morano [29] proposed an algorithm for constructing an M-array, fixing the length of the alphabet, the window property size, the dimensions of the array and the Hamming distance between windows. The algorithm used to generate an array with fixed properties is based on a brute force approach. In our case, when constructing a binary M-array with window property of 5×5, the following steps are taken: firstly, a sub-array of 5×5 is chosen randomly and is placed to the top-left corner of the M-array that is to be built. Then consecutive random columns of 5×1 are added to the right of this initial sub-array, maintaining the integrity of the window property of the array. Afterward, rows of 1×5 are added beneath the initial sub-array in a similar way. Then both the horizontal and vertical processes are repeated by incrementing the starting coordinates by one, until the whole array is filled up. When filling the array, the code uniqueness of each newly added point is checked. If it is not satisfied, the array is cleared and the algorithm starts again. Since the computational complexity is extremely high, the author only generated an array of 45×45. Besides the three aforementioned methods, some other typical methods in binary spatial coding are listed in Table I. 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 paper we seek to generate the pattern array using statistical analysis.

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 the 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 an $(m \times n)$-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.

Fig. 4. Magnified part of the binary pattern (the dotted line grid is added for illustration).

In our experimentation, after conducting 100 trials of pattern generation, the array with the largest average intercodeword Hamming distance ($\bar{H} = 4.524$) was derived. The large intercodeword Hamming distance corresponds to good noise-tolerance of the codewords on the imaging side. We chose this array (part of which is shown in Fig. 4) to embed into arbitrary 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.

VI. 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 projector-camera system.

A. Hand Segmentation

Hand segmentation, as the first step for most barehand-based applications, plays an important role in the robustness, accuracy and efficiency of an HCI system. The approaches for hand segmentation have been studied extensively in the computer vision community.

Among them, skin color detection [30], [31] is very common for its simplicity and ease of implementation, and is very efficient against the case of simple background or hand being the only skin-colored object. However, in the projector-camera scenario we have, diverse video contents are projected continuously. When some skin-colored objects could be projected onto the background (Region A in Fig. 5), or non-skin-colored objects are projected onto the hand (Region B in Fig. 5),

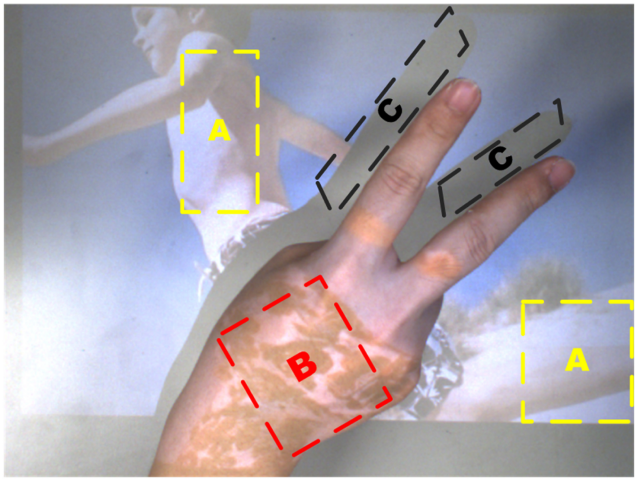


Fig. 5. Sample hand image captured by projector-camera system.

the effectiveness of the skin color based methods could be compromised severely.

Since the geometrically and radiometrically calibrated projector-camera system allows to predict where the video contents are projected and how they should appear in the image data, background subtraction [32] could be adopted to segment the hand as the set of pixels that are out of expectation on the projection surface. However, the approach generally could not separate the hand region from the hand-cast shadows (Region C in Fig. 5). It is also generally sensitive to the quality of the calibration procedures, and demands constant ambient illuminations and fixed projection surface.

The graph-based [33], [34] approaches are generally able to generate good segmentations. However, the time-consuming nature of these approaches and the demand on user interaction weaken their advantage for the HCI application in which speed is an important factor for real-time interaction.

Instead of using monocular camera, some researchers use additional instruments, such as infrared camera [35], stereo cameras [36], depth sensor [37], and so on, to distinguish the hand region from the background. However, the additional hardware inevitably increases the complexity of the projector-camera system configuration.

In our system, we adopt a coarse-to-fine approach to solve the aforementioned problems. The main idea is to combine contrast saliency map with mean-shift based smoothing and segmentation via a confidence function. Low-level contrast saliency detection enables the hand region to be highlighted coarsely, and mean-shift based smoothing method removes the noises induced by the arbitrary projection contents without demolishing the discontinuity information. Moreover, even without demanding pretraining and precalibration procedures, the approach still allows robust, precise, and rapid hand segmentation to be achieved.

1) *Coarse Segmentation by Contrast Saliency:* Despite the presence of incessant varied video contents projected to the projection surface and the hand operating above the surface, the hand is almost always the most noticeable object to human vision. Motivated by this, we firstly employ a saliency detector to reach a coarse hand region segmentation. Salient region detection as a typical low-level vision approach has been widely studied in computer vision. According to the specific projector-camera scenario we have, the saliency detector must satisfy the following requirements:

1) emphasize the largest salient objects;
2) uniformly highlight the whole salient regions;
3) disregard artifacts arising from the projection content and ambient illumination;
4) accomplish detection in less than $15ms$ to meet the real-time requirement.

Upon comparing different saliency detection methods [38]–[40], we chose the histogram-based contrast [41] method, which best fulfills the aforementioned criteria, to define the saliency values for image pixels.

The saliency of a pixel is defined using its color contrast with respect to all other pixels in the image, i.e., the saliency value of a pixel I_k in image I is defined as

$$S(I_k) = \sum_{i=1}^{N} D(I_k, I_i) \tag{8}$$

where $D(I_k, I_i)$ is the color distance metric between pixels I_k and I_i in the HSV color space. It is clear that pixels with the same color value have the same saliency value under the definition, since the measure is oblivious to spatial relations. Hence, rearranging (8) so that the terms with the same color value c_j are grouped together, we get the saliency value for each color as

$$S(I_k) = S(c_l) = \sum_{j=1}^{n} f_j D(c_l, c_j) \tag{9}$$

where c_l is the color value of pixel I_k, n is the number of distinct pixel colors, and f_j is the probability of having pixel color c_j in image I.

To reduce the high dimensionality of the 256^3 true-color space, the more frequently-emerging 85 colors were selected by building a compact color histogram using color quantization. With that, artifacts would be introduced. A smoothing procedure is used to refine the saliency value of each color, which replaces it with the weighted average of the saliency value of similar colors. Typically, $m = n/4$ nearest colors are chosen to refine the saliency value of color c by

$$S'(c) = \frac{1}{(m-1)T} \sum_{i=1}^{m} [T - D(c, c_i)] S(c_i) \tag{10}$$

where $T = \sum_{i=1}^{m} D(c, c_i)$ is the sum of the distances between color c and its m nearest neighbors c_i, and the normalization factor comes from

$$\sum_{i=1}^{m} [T - D(c, c_i)] = (m-1)T. \tag{11}$$

More implementation issues are detailed in [41]. The saliency map $S(x, y)$ of image I [Fig. 6(a)] is derived as illustrated in Fig. 6(b).

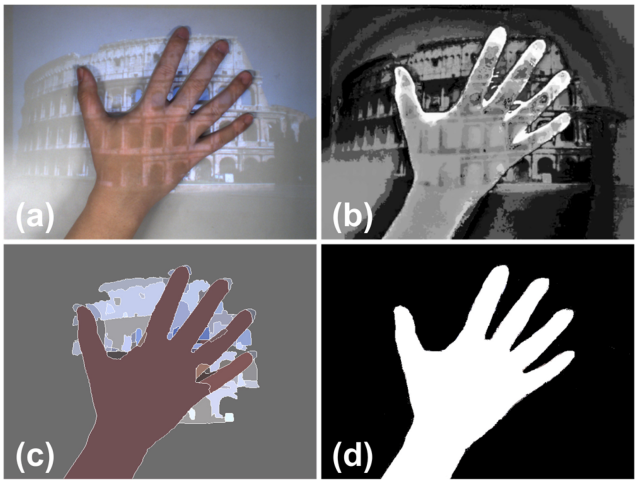


Fig. 6. Hand segmentation. (a) Original image. (b) Histogram contrast salient map. (c) Segments derived through mean-shift. (d) Refined segmentation result.

2) *Mean-Shift Region Smoothing:* Even though the hand region has been highlighted through saliency detection, as illustrated in Fig. 6(b), it is not uniformly emphasized due to the influence of the projection content on the hand and the projection surface. It is generally impossible to have precise hand segmentation through traditional threshold-based methods. To tackle the issue, we apply the mean-shift based smoothing and segmentation approach [42] to the salient regions, which not only eliminates the noises but also preserves the discontinuity by adaptively reducing the amount of smoothing near the abrupt changes in the local structure, i.e., over the boundaries.

One important advantage of mean shift-based segmentation is its capability to resolve the over-segmentation issue. The joint-domain mean shift-based segmentation succeeds in overcoming the inherent limitations of the methods based only on gray-level or color clustering which typically over-segment small gradient regions. Notice that small intensity-gradient regions are not uncommon in the projector-illuminated area due to the projector's nonlinearity and the variations of ambient illuminations.

Another important advantage of mean shift-based segmentation [42] is its modularity which makes the control of segmentation output simple. The control is just through three parameters: (h_s, h_r, M). The range parameter h_r and the smallest significant feature size M control the number of regions in the segmented piecewise constant model; generally larger values have to be used for h_r and M to discard the effect of small local variation. The spatial parameter h_s determines the size of the spatial window. In our case, (h_s, h_r, M) is set to $(7, 10, 20)$.

It is worth mentioning that the inherent iterative nature of the mean-shift based method often invokes the efficiency problem. However, the coarse salient region detection conducted earlier could reduce the mean-shift search space and accelerate the convergence speed dramatically.

After mean-shift smoothing and segmentation, the image is divided into L candidate partitions P_k, $i = 1, \ldots, L$, as shown in Fig. 6(c). The contour of the hand is generally preserved well.

3) *Precise Segmentation by Fusing:* To acquire precise hand region segmentation, we propose a confidence function that puts contrast saliency and region discontinuity together to evaluate the probability of a candidate partition for being a part of the hand region. The value of confidence function for each candidate partition is determined by several terms as listed below:

1) the average salient value of the pixels in the partition;
2) the number of the neighbor partitions and the average salient value of neighbor partitions;
3) the area of the partition;
4) whether the partition is on the image boundary.

Hence, the value of the confidence function $C_F(k)$ for partition P_k is formulated as

$$C_F(k) = \frac{1}{e^{(L-1)}}[\alpha \bar{S}(k) + \beta \bar{S}_N(k) + \gamma A(k)] \tag{12}$$

where $\bar{S}(k)$ is the average saliency value of the pixels in P_k, $\bar{S}_N(k)$ is the average saliency value of its N neighbor partitions, and $A(k)$ is the partition's area. The three terms above are all scaled to $[0, 1]$. L is the number of image boundary segments to which the partition attached; $L \geq 2$ indicates that the partition belongs to background that should have low confidence value. The weights are α, β, γ. If one partition is an isolated area in the hand region or background region, the confidence value of that partition would depend mostly on its surrounding neighborhood. Hence, when the number of neighbor partitions N is 1, $\beta = 1/2, \alpha = \gamma = 1/4$; otherwise, $\alpha = 1/2, \beta = \gamma = 1/4$.

If $C_F(k)$ is greater than a predefined threshold Δ, the partition is considered as a part of the hand region. Since not all skin pixels are categorized always correctly, a morphological closing operation is employed to remove small noisy holes in the skin pixel areas. This way, the refined binary segmentation is reached, as shown in Fig. 6(d).

B. Fingertip Detection

Fingertip detection is conducted on the basis of the segmented binary hand image. As illustrated in Fig. 7(b), the hand contour is retrieved from the binary image using the algorithm detailed in [43], The extracted contour serves to offer fingertip candidates through a simple arc line analysis. Let $\mathbf{T}(x), x = 1, \ldots, N$ 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 as the angle θ

$$\theta = \arccos \frac{\mathbf{v_1} \cdot \mathbf{v_2}}{\|\mathbf{v_1}\| \|\mathbf{v_2}\|} \tag{13}$$

$$\mathbf{v_1} = \mathbf{T}(k) - \mathbf{T}(k - t) \tag{14}$$

$$\mathbf{v_2} = \mathbf{T}(k) - \mathbf{T}(k + t) \tag{15}$$

where $\mathbf{T}(k - t)$ and $\mathbf{T}(k + t)$ are contour points in the vicinity of $\mathbf{T}(k)$, each on a different side of $\mathbf{T}(k)$ at an interval of t points from it.

If $\theta < \frac{\pi}{2}$ and $|\mathbf{v_1}, \mathbf{v_2}| > 0$, $\mathbf{T}(k)$ is regarded as a fingertip candidate. The second conditional term as a determinant is employed to distinguish the fingertip peaks from the valleys

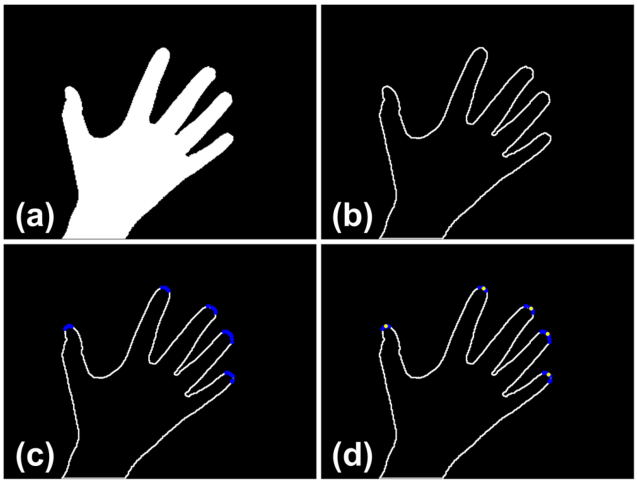


Fig. 7. Fingertip detection. (a) Binary hand image. (b) Hand contour. (c) Fingertip candidates. (d) Detected fingertips.

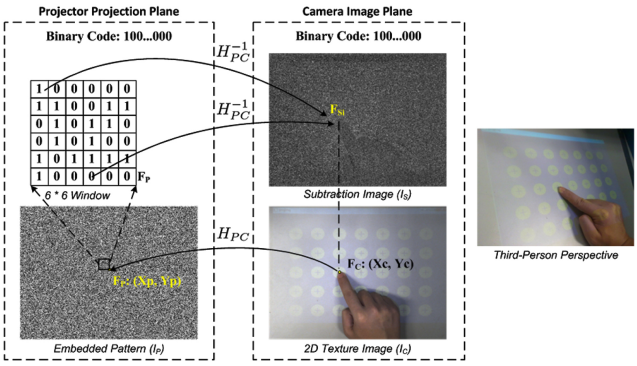


Fig. 8. Touch detection via homography.

between fingers. Some fingertip candidates, indicated by blue points in Fig. 7(c), are thus extracted. Finally, the candidates that are consecutive or nearly consecutive in the hand silhouette are clustered into the same group, and in each group only the candidate in the median position is regarded as a fingertip [yellow points in Fig. 7(d)].

C. 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 generally infeasible to align pixels on the camera's image plane to 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. Multitouch detection is an extension of single-touch detection.

As illustrated in Fig. 8, 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

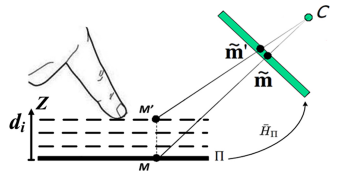


Fig. 9. Homography transfer across parallel planes.

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

$$BC_P = \sum_{i=0}^{35} 2^i \cdot I_P(F_{P_i}) \tag{16}$$

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

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

$$BC_S = \sum_{i=0}^{35} 2^i \cdot I_S(F_{S_i}) \tag{17}$$

$$\tilde{F}_{S_i} = H_{PC}^{-1}\tilde{F}_{P_i} \tag{18}$$

where $\tilde{F}_{S_i}$ and $\tilde{F}_{P_i}$ are the 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 3-D reconstruction, and can operate in real-time.

D. From Resistive Touching to Capacitive Touching

In the last subsection, we have emulated a resistive touch operation, which requires touching with a certain pressure on the projection surface. Below we show how to enhance the touch sensitivity and adjust the interface from a resistive-like touch to a capacitive-like 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. 9. 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 is regarded as confirmed.

As shown in Fig. 9, 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 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 precalibrated projection matrix of the camera be

$$P \cong [p_1, p_2, p_3, p_4] \cong K[r_{1\Pi}, r_{2\Pi}, r_{3\Pi}, t_\Pi] \tag{19}$$

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

$$\tilde{m} \cong \bar{H}_\Pi [X, Y, 1]^T \tag{20}$$

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

Suppose we have a plane Π_{d_i} parallel to but elevated from Π by a perpendicular distance d_i. For the 3-D 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

$$\begin{aligned} \tilde{m}' &\cong K[R_\Pi, t_\Pi][X, Y, d_i, 1]^T \\ &\cong K(Xr_{1\Pi} + Yr_{2\Pi} + d_i r_{3\Pi} + t_\Pi) \\ &\cong K([r_{1\Pi}, r_{2\Pi}, t_\Pi] + d_i[0, 0, r_{3\Pi}])[X, Y, 1]^T \\ &\cong (\bar{H}_\Pi + d_i[0, 0, Kr_{3\Pi}])[X, Y, 1]^T. \end{aligned} \tag{21}$$

By substituting (20) into (21), we have

$$\tilde{m}' \cong (I + d_i[0, 0, p_3]\bar{H}_\Pi^{-1})\tilde{m} \cong H_{Cd_i}\tilde{m}. \tag{22}$$

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 the 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 the 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.

VII. Experiments

To assess the feasibility of the described system for bare-hand human-computer interface, we conducted experiments to evaluate display quality, hand segmentation accuracy, touch detection accuracy, trajectory tracking accuracy, multitouch capability, and system efficiency respectively.

A. System Initialization

For any camera-projector-table system, projection keystone correction is a necessary process, which in our case is accomplished by the use of the homography between the projector's display panel and the table surface. On the other hand, the finger touch action is determined through the homography between the camera's image plane and the projector's dipslay panel, that is induced by the planar table surface. Therefore, before the system's operation, the initialization step is to estimate the camera-table and camera-projector homoraphies.

1) *Camera-Projector Homography Estimation:* To estimate the camera-projector homography, one projection-capture cycle is needed. As shown in Fig. 10(a), a chessboard pattern was projected onto the table surface, the chessboard corners $CP_i (i = 0, \ldots, N)$ indicated by the blue circle were considered as the feature points, and the coordinates of these points were known from the chessboard generation process. Using camera, an image of the table surface illuminated by the chessboard was acquired. Then through the automatic corner detection in the image data, the corresponding points $CC_i (i = 0, \ldots, N)$ were found, as indicated by the white dots in Fig. 10(b). Finally, by the use of $CP_i \sim CC_i$ correspondences, the camera-projector homography H_{CP} can be calculated by the least-square method.

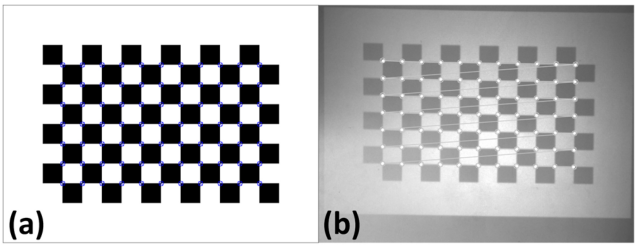


Fig. 10. Images for camera-projector homography estimation. (a) Projected chessboard. (b) Captured image.

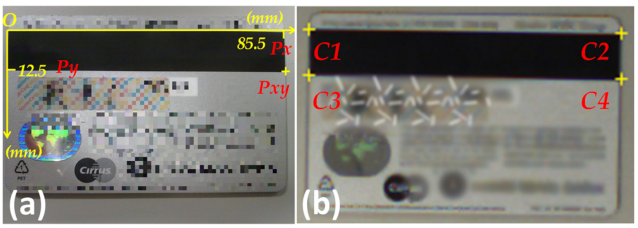


Fig. 11. Images for camera-table homography estimation. (a) Credit card. (b) Captured image.

2) *Camera-Table Homography Estimation:* Because of the lack of sensing capability of the projector, it is impossible to estimate the projector-table homography directly. However, With the camera-projector homography already obtained, plus knowledge of the camera-table homography, the projector-table homography can be determined through homography composition. To deetermine the camera-table homography, a planar object with standard known dimension is required. As illustrated in Fig. 11(a), a credit card was employed as the calibration object, the black magnetic stripe on it was a rectangle with standard dimension of $85.5mm \times 12.5mm (W \times H)$. The top-left corner was chosen as the origin of the coordinate system of the card, and thus the coordinates of the four corners O, P_x, P_y and P_{xy} were $(0, 0)$, $(85.5, 0)$, $(0, 12.5)$ and $(85.5, 12.5)$ respectively. The credit card was put onto the table surface, and then an image was captured as shown in Fig. 11(b). After binary segmentation and corner detection, four corresponding points $C_i, i = 1, \ldots, 4$ were detected in the image, as indicated by the yellow crosses in Fig. 11(b). Then through the four correspondences ($C_1 \sim O$, $C_2 \sim P_x$, $C_3 \sim P_y$, $C_4 \sim P_{xy}$), the camera-table homograpy could be determined.

Thus the initialization step requires only one image to be projected and two images to be captured. Including the computation time, the initialization can be accomplished within five seconds.

B. 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

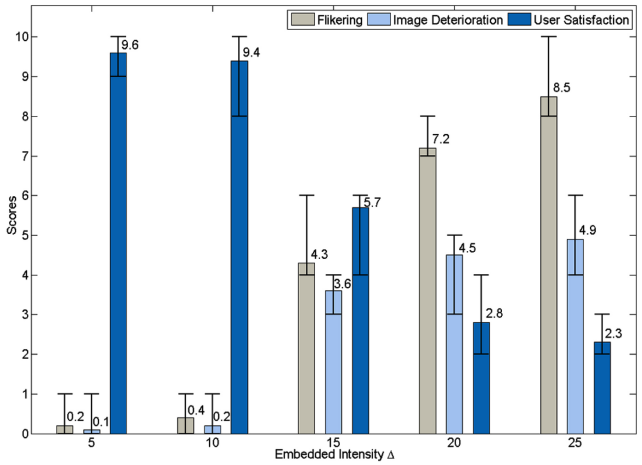


Fig. 12. User study results on code imperceptibility.

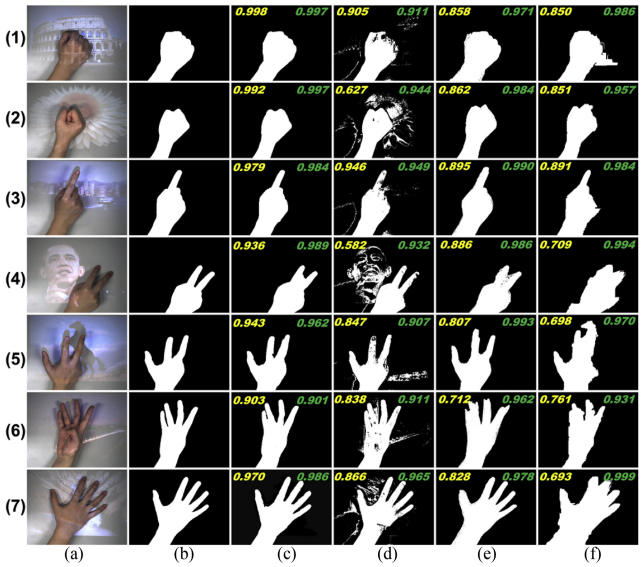


Fig. 13. Visual comparison. (a) Original image. (b) Ground-truth. (c) Our method. (d) SCM [30]. (e) BkSub [32]. (f) GB [33]. The yellow (top-left) and green (top-right) numbers in each result image are the corresponding precision p and recall r values, respectively.

invited to participate in this experiment. 500 images were collected from Google Image randomly, to which binary pattern was embedded with different intensities. The viewers were seated in front of a desk surface to which the video contents were projected, and asked to comment on the quality of the projections. The questions asked were simplified from the questionnaire used in [44], focusing on the feeling of flickering, the recognition of image deterioration, and the overall satisfaction on the projection quality. The score for each question ranged from 0 to 10.

The average scores of the subjective evaluation are illustrated in Fig. 12. In practice, because of the limited projection intensity of the consumable-grade projector we used in our experimentation, we chose $\Delta = 10$ in our configuration to strike a compromise between user satisfaction and code imperceptibility.

C. Hand Segmentation Accuracy Evaluation

As the first step of touch detection, how accurately the hand region is segmented has a direct impact to the performance

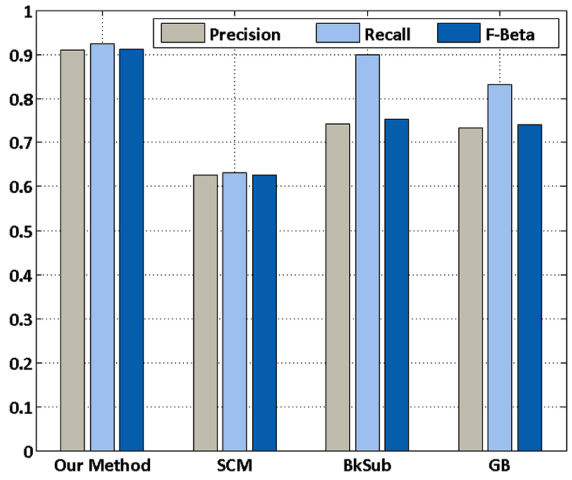


Fig. 14. Precision-recall bars for hand segmentation using different methods. Our method shows high precision, recall and F_β values.

TABLE II
QUANTITATIVE EXPERIMENTAL RESULTS

Surface	Illumination			
	Dark		Normal	
	ϵ(px)	FRR/FAR(%)	ϵ(px)	FRR/FAR(%)
Gray	2.54	1.11/0.43	2.84	1.30/0.46
Yellow	2.87	1.23/0.55	2.95	1.49/0.59
Artifact	3.02	1.75/0.64	3.17	1.66/0.58

of touch detection. It is thus necessary to evaluate first the accuracy of hand segmentation.

We collected a great diversity of images (e.g., flowers, buildings, celebrities, animals etc.) from Google Image and projected them to a desk surface. An experimental dataset of 500 images was captured under different projection contents and different hand shapes. The ground-truth was manually annotated with the assistance of GrabCut [34]. Several test images with their ground-truth are shown in Fig. 13(a) and (b).

To illustrate the merits of the proposed method, we conducted comparison experiments with some related methods. The choice of these methods is motivated by the following reasons: citation count in the literature (the classical approach of statistical color model-based (SCM) method is widely cited [30]), precision [the background subtraction method (BkSub) has higher precision, since it is on the basis of using precalibrated geometric and radiometric information to predict the background image [32]], and recency [the sophisticated graph based method (GB) [33] is one of the latest methods].

As in [41], we adopted the F-beta score to evaluate the accuracy of segmentation, which considers both the precision p and the recall r to compute the score: $p = N_C/N_R$, $r = N_C/N_G$, where N_C, N_R, N_G are the numbers of correctly segmented pixels, all segmented pixels and ground-truth pixels respectively. The F-beta score is the harmonic mean of precision and recall, formulated as

$$F_\beta = (1+\beta^2)\cdot\frac{\mathrm{p}\cdot\mathrm{r}}{(\beta^2\cdot\mathrm{p})+\mathrm{r}} \tag{23}$$

where β is set to 0.3 to weigh precision more than recall [41], [39]. The visual and quantitative comparisons are shown in

TABLE III
COMPARISON WITH PREVIOUS SYSTEM IN DIFFERENT SCENARIOS

Scenario[a]	Method			
	Proposed system[b]		PROCAMS12 [1]	
	ϵ(px)	FRR/FAR(%)	ϵ(px)	FRR/FAR(%)
$(S_G + L_N)$	2.84	1.30/0.46	3.05	1.32/0.48
$(S_G + L_N) \rightarrow (S_G + L_D)$ w/ C_P	2.54	1.11/0.43	2.98	1.12/0.45
$(S_G + L_N) \rightarrow (S_G + L_D)$ w/o C_P			10.67	50.57/25.96
$(S_G + L_N) \rightarrow (S_A + L_N)$ w/ C_P	3.17	1.66/0.58	3.20	1.76/0.63
$(S_G + L_N) \rightarrow (S_A + L_N)$ w/o C_P			15.73	52.65/29.14

[a] Term abbrev.: gray surface (S_G), artifact surface (S_A), normal illumination (L_N), dark illumination (L_D), photometric calibration (C_P).
[b] Photometric calibration doesn't affect the performance of proposed system.

Figs. 13 and 14 respectively. Among all the methods, our method shows the highest precision, recall and F_β values. It is expected that the skin color-based method (SCM) gets low precision when some projected objects have color similar to that of the skin; examples of such objects are human face and yellow flower as shown in Fig. 13(d2) and (d4). The background subtraction method (BkSub) shows a high recall but poor precision, verifying that the shadow cast by video projection has great influence to the method, as shown in Fig. 13(e4) and (e6). The graph-based method (GB) cannot preserve smooth boundaries and often confuse the projected objects with the hand region, which are the main reasons for low precision score, as illustrated in Fig. 13(f4)–(f7).

D. Touch Accuracy Evaluation

Similar to [21], we specifically designed an image, in which 35 circles were distributed uniformly. As shown in Fig. 15(a), 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. 15(b)–(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. 15(b) and (d)]. Five persons were invited to participate in the experiment, each of them conducted six rounds (on the three surfaces and under two ambient illuminations). Totally, 1050 touch trials were produced.

The precision of touch position localization is evaluated by the average distance between ground-truth and the detected position, which is formulated as

$$\epsilon = \frac{1}{N_t} \sum_{i=1}^{N_t} \sqrt{(X_{d_i} - X_{g_i})^2 + (Y_{d_i} - Y_{g_i})^2} \tag{24}$$

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.

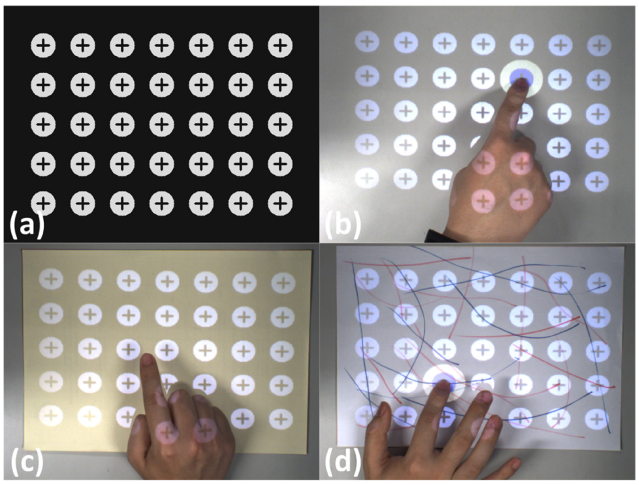


Fig. 15. (a) Image projected for ground-truth collection. (b) Gray surface. (c) Yellow surface. (d) Surface with artifacts.

The detailed quantitative testing results, listed in Table II, 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 [20], 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 OmniTouch [21], 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 exactly identical, the comparison results show that the described system has at least comparable performance even with the use of simpler devices. Some frames from one trial are shown in Fig. 16. There the camera view, third person view, and fingertip trajectory are also demonstrated in each sub-figure.

Furthermore, we compared touch detection accuracy between proposed system and previous *PROCAMS12* system [1] in different scenarios, the detailed results are shown in Table III. The performance in scenario with gray projection surface and dark ambient lighting is set as benchmark. Compared with the background subtraction method in hand segmentation module of *PROCAMS12* system, the new hand segmentation method has higher accuracy, as evaluated

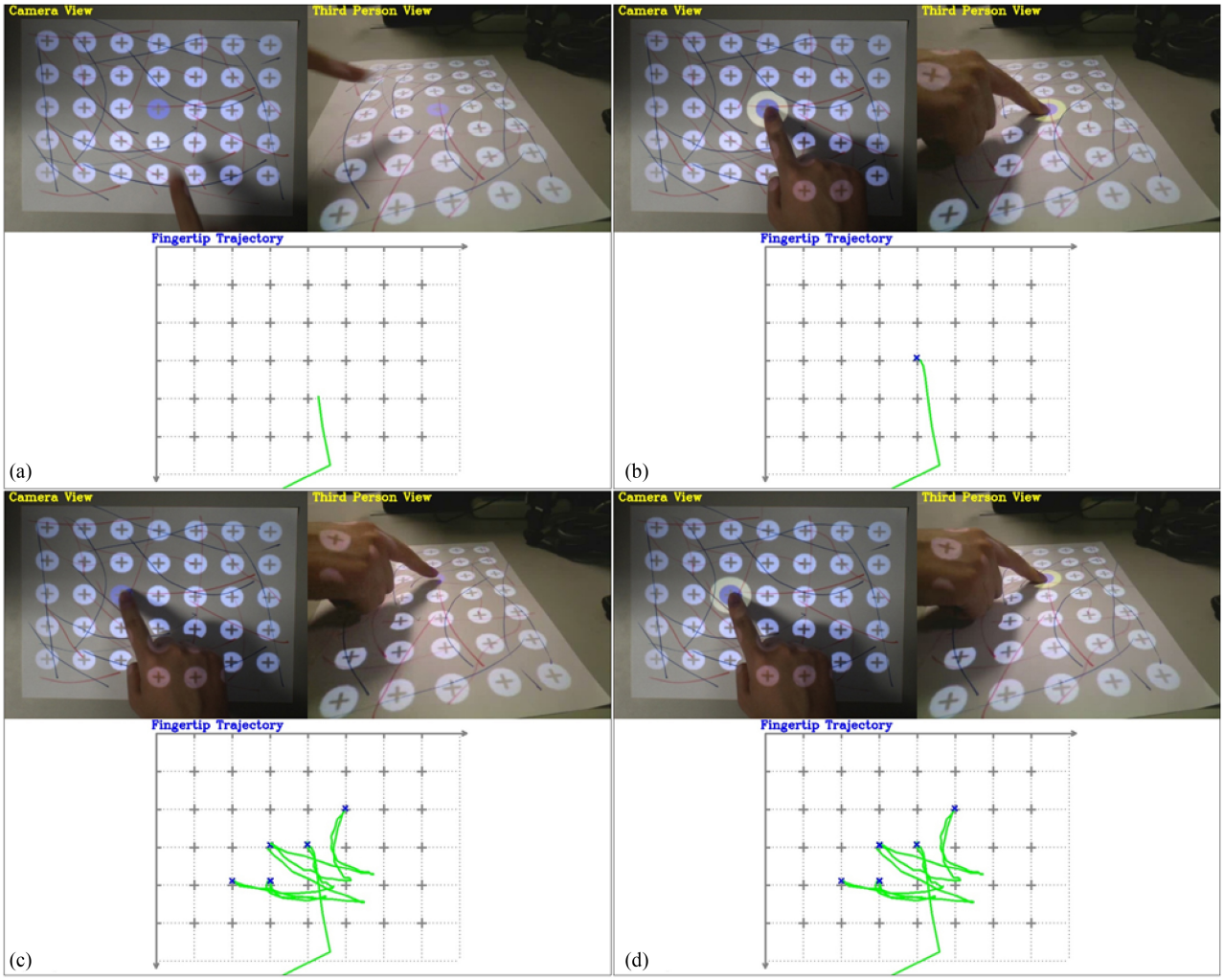


Fig. 16. Some frames from one trial (regular ambient illumination and artifacts surface) on touch accuracy evaluation. (a) Frame 1019. (b) Frame 1032. (c) Frame 1192. (d) Frame 1198.

in Section VII-C, therefore the proposed system has better touch detection performance. What's more, since background subtraction method greatly depends on accurate background prediction, if projection surface or surrounding illumination is changed, the *PROCAMS12* system has to be recalibrated to update the radiometric parameters. Otherwise, the performance will have a remarkable degradation. As demonstrated in the scenarios of changing to artifact surface or normal illumination without radiometric calibration, more than 75% touch actions cannot be detected correctly. This disadvantage makes previous system unadaptable for mobile applications in which projection surface and surrounding lighting always change.

E. Trajectory Tracking Evaluation

Besides clicking, finger dragging is also an important action in typical touch screen operation. We conducted an evaluation of trajectory tracking when finger was dragged on the projection surface. As shown in Fig. 17(a), three different geometrical shapes (square, right triangle and circle) were projected onto the table surface. Five users were asked to drag their index finger along three boundaries one by one. The average trajectories indicated by blue curves [as shown in Fig. 17(b)] almost coincided with the ground-truth in gray. This experiment shows that our method can track the trajectory of dragged finger precisely.

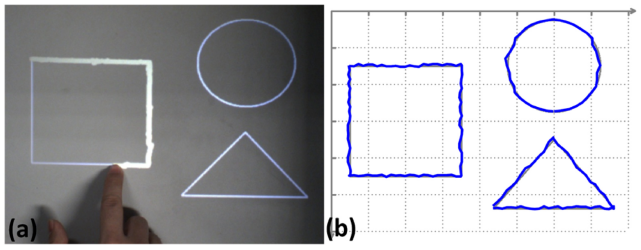


Fig. 17. (a) Image projected for ground-truth collection. (b) Fingertip dragging trajectories.

F. Multiple-Touch Evaluation

Multitouch refers to a touch sensing surface's ability to recognize the presence of two or more points of contact with the surface. This plural-point awareness is often used to implement advanced functionality such as pinch to zoom or activating predefined programs. In the aforementioned

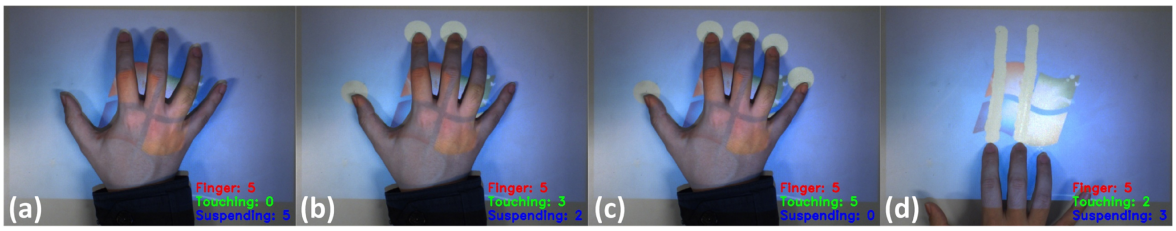


Fig. 18. Some frames from one trial on multitouch capability evaluation. (a) Frame 10641. (b) Frame 10809. (c) Frame 10813. (d) Frame 11601.

TABLE IV
AVERAGE PROCESSING TIME

Method	Subroutine Time (ms)				
	Calib.	Hand Seg.	FTip Loc.	Touch Det.	Total
Ours	122.31	23.15	1.31	1.74	26.20
[1]	1954.84	14.63	1.32	1.74	17.69

experiments, our method has been shown as an accurate and effective method for tracking the state of a single finger. Our system is amenable in that single-touch sensing can be easily extended to multitouch sensing. Some key frames from one trial are demonstrated in Fig. 18, revealing the feasibility on the multitouch case.

G. Efficiency Evaluation

For human-computer interface, real-time performance is of great importance. We implemented the proposed system in C++ using the Intel OpenCV Library to speed up the processing time. Through multithread 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 IV shows the average processing times for off-line calibration, segmentation in the image domain, fingertip localization, and touch detection in proposed system and previous system [1] respectively. The time consumed by off-line calibration does not include projection-capture cycles for calibration data collection. Because proposed system adopts novel hand segmentation approach which does not require photometric calibration, the computation time is much shorter than that of previous system. Moreover, the proposed system just needs one projection and two capture operations, the actual calibration time is even shorter than that of previous system, which needs six projection and seven capture operations. Among all on-line subroutines, hand segmentation spends more time due to the iterative characteristic of mean-shift algorithm. The total time consumption is less than 30ms, indicating the system meets the requirement of real-time application. Although the total time consumption of online calculation is greater than that of previous system, the less time consuming in calibration will provide more flexibility in mobile applications.

VIII. 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.

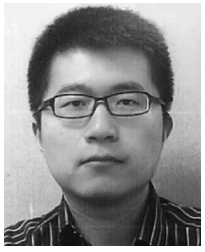
The described work lays down the setup and design of the projector-camera system for touch-sensitive interface. Single-touch, touch dragging tracking and multitouch facilities are also constructed and thoroughly experimented with. All these form the basis of a more complete touch interface system.

Future work includes more thorough experimentation with multihand interface using the system. Based upon the touch detection facility, advanced touch gestures (e.g., double click, scroll, zoom-in, zoom-out) and even typing recognition on the described platform will be studied. It is necessary to transplant the algorithm to mobile devices which have limited computing resource, although our method can run at about 50 frames/s on a desktop computer, it is not fast enough for embedded system. The hand segmentation subroutine spends most of the time, more effective hand segmentation approach will be investigated. Moreover, the diverse anatomy of a human finger allows different parts of it to be recognized—including the tip, pad, nail and knuckle—without having to instrument the user. This opens several new and powerful interaction opportunities for touch input. Unlike TapSense [45] that demands the use of additional acoustic sensor to classify unique signatures that different objects create when striking a touch surface, in our system some visual recognition methods could be explored to see if different parts of a finger touching a table surface can be identified. This is another direction of our future work.

REFERENCES

[1] J. Dai and R. Chung, "Making any planar surface into a touch-sensitive display by a mere projector and camera," in *Proc. IEEE PROCAMS*, 2012, pp. 35–42.
[2] J. Patten, H. Ishii, J. Hines, and G. Pangaro, "Sensetable: A wireless object tracking platform for tangible user interfaces," in *Proc. ACM CHI*, 2001, pp. 253–260.
[3] E. D. Mynatt, T. Igarashi, W. K. Edwards, and A. LaMarca, "Flatland: New dimensions in office whiteboards," in *Proc. ACM CHI*, 1999, pp. 346–353.
[4] M. Ashdown and P. Robinson, "Escritoire: A personal projected display," *IEEE MultiMedia Mag.*, vol. 12, no. 1, pp. 34–42, Jan.–Mar. 2005.
[5] B. Jones, R. Sodhi, R. Campbell, G. Garnett, and B. Bailey, "Build your world and play in it: Interacting with surface particles on complex objects," in *Proc. IEEE ISMAR*, 2010, pp. 165–174.
[6] P. Dietz and D. Leigh, "DiamondTouch: A multi-user touch technology," in *Proc. ACM UIST*, 2001, pp. 219–226.
[7] J. Rekimoto, "Smartskin: An infrastructure for freehand manipulation on interactive surfaces," in *Proc. ACM CHI*, 2002, pp. 113–120.
[8] Light Blue Optics. *Light Touch* [Online]. Available: http://lightblueoptics.com.
[9] S. K. Kane *et. al*, "Bonfire: A nomadic system for hybrid laptop-tabletop interaction," in *Proc. ACM UIST*, 2009, pp. 129–138.

[10] C. Harrison, D. Tan, and D. Morris, "Skinput: Appropriating the body as an input surface," in *Proc. ACM CHI*, 2010, pp. 453–462.
[11] I. Katz, K. Gabayan, and H. Aghajan, "A multitouch surface using multiple cameras," in *Advanced Concepts for Intelligent Vision Systems, LNCS*, vol. 4678. Berlin, Germany: Springer, 2007, pp. 97–108.
[12] J. Letessier and F. Bérard, "Visual tracking of bare fingers for interactive surfaces," in *Proc. ACM UIST*, 2004, pp. 119–122.
[13] C. von Hardenberg and F. Bérard, "Bare-hand human-computer interaction," in *Proc. Workshop Perceptive User Interfaces*, 2001, pp. 1–8.
[14] R. Kjeldsen, C. Pinhanez, G. Pingali, J. Hartman, T. Levas, and M. Podlaseck, "Interacting with steerable projected displays," in *Proc. IEEE FG*, 2002, pp. 402–407.
[15] J. Marshall, T. Pridmore, M. Pound, S. Benford, and B. Koleva, "Pressing the flesh: Sensing multiple touch and finger pressure on arbitrary surfaces," in *Proc. Int. Conf. Pervasive Comput.*, 2008, pp. 38–55.
[16] P. Song, S. Winkler, S. O. Gilani, and Z. Zhou, "Vision-based projected tabletop interface for finger interactions," in *Proc. IEEE HCI*, 2007, pp. 49–58.
[17] A. D. Wilson, "PlayAnywhere: A compact interactive tabletop projection-vision system," in *Proc. ACM UIST*, 2005, pp. 83–92.
[18] Fitriani and W.-B. Goh, "Interacting with projected media on deformable surfaces," in *Proc. IEEE ICCV*, 2007, pp. 1–6.
[19] A. D. Wilson and H. Benko, "Combining multiple depth cameras and projectors for interactions on, above and between surfaces," in *Proc. ACM UIST*, 2010, pp. 273–282.
[20] A. D. Wilson, "Using a depth camera as a touch sensor," in *Proc. ACM Int. Conf. Interact. Tabletops Surfaces*, 2010, pp. 69–72.
[21] C. Harrison, H. Benko, and A. D. Wilson, "Omnitouch: Wearable multitouch interaction everywhere," in *Proc. ACM UIST*, 2011, pp. 441–450.
[22] Z. Song and R. Chung, "Use of LCD panel for calibrating structured-light-based range sensing system," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 11, pp. 2623–2630, Nov. 2008.
[23] R. Hartley, and A. Zisserman, *Multiple View Geometry in Computer Vision(2e)*. London, U.K.: Cambridge Univ. Press, 2004.
[24] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs, "The office of the future: A unified approach to image-based modeling and spatially immersive displays," in *Proc. ACM SIGGRAPH*, 1998, pp. 179–188.
[25] A. Grundhöfer, M. Seeger, F. Hantsch, and O. Bimber, "Dynamic adaptation of projected imperceptible codes," in *Proc. IEEE/ACM ISMAR*, 2007, pp. 1–10.
[26] J. Dai and C. Chung, "Embedding invisible codes into normal video projection: Principle, evaluation and applications," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.
[27] F. MacWilliams and N. Sloane, "Pseudo-random sequences and arrays," *Proc. IEEE*, vol. 64, no. 12, pp. 1715–1729, Dec. 1976.
[28] T. Etzion, "Constructions for perfect maps and pseudorandom arrays," *IEEE Trans. Inf. Theory*, vol. 34, no. 5, pp. 1308–1316, Sep. 1988.
[29] R. Morano, C. Ozturk, R. Conn, S. Dubin, S. Zietz, and J. Nissano, "Structured light using pseudorandom codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 322–327, Mar. 1998.
[30] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *Int. J. Comput. Vision*, vol. 46, no. 1, pp. 81–96, 2002.
[31] M. Donoser and H. Bischof, "Real time appearance based hand tracking," in *Proc. ICPR*, 2008, pp. 1–4.
[32] A. Licsár and T. Szirányi, "Hand gesture recognition in camera-projector system," in *Proc. ECCV Workshop HCI*, 2004, pp. 83–93.
[33] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.
[34] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.
[35] Y. Sato, Y. Kobayashi, and H. Koike, "Fast tracking of hands and fingertips in infrared images for augmented desk interface," in *Proc. IEEE FG*, 2000, pp. 462–467.
[36] Q. Wang, X. Chen, and W. Gao, "Skin color weighted disparity competition for hand segmentation from stereo camera," in *Proc. BMVC*, 2010, pp. 1–11.
[37] M. Van den Bergh and L. Van Gool, "Combining RGB and ToF cameras for real-time 3-D hand gesture interaction," in *Proc. IEEE WACV*, 2011, pp. 66–72.
[38] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE CVPR*, 2007, pp. 1–8.
[39] R. Achanta, S. S. Hemami, F. J. Estrada, and S. Süsstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE CVPR*, 2009, pp. 1597–1604.
[40] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," in *Proc. IEEE CVPR*, 2010, pp. 2376–2383.
[41] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *Proc. IEEE CVPR*, 2011, pp. 409–416.
[42] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[43] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Comput. Vision, Graph. Image Process.*, vol. 30, no. 1, pp. 32–46, 1985.
[44] H. Park, B.-K. Seo, and J.-I. Park, "Subjective evaluation on visual perceptibility of embedding complementary patterns for nonintrusive projection-based augmented reality," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 5, pp. 687–696, May 2010.
[45] C. Harrison, J. Schwarz, and S. E. Hudson, "Tapsense: Enhancing finger interaction on touch surfaces," in *Proc. ACM UIST*, 2011, pp. 627–636.



Jingwen Dai (S'09–M'12) received the B.E. degree in automation from Southeast University, Nanjing, China, in 2005, the M.E. degree in automation from Shanghai Jiao Tong University, Shanghai, China, in 2009, and the Ph.D. degree in mechanical and automation engineering from the Chinese University of Hong Kong, Hong Kong, in 2012.

He is currently a Post-Doctoral Research Associate with the Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA. His current research interests include computer vision and human-computer interaction.

Chi-Kit Ronald Chung (SM'99) received the B.S.E.E. degree from the University of Hong Kong, Hong Kong, and the Ph.D. degree in computer engineering from the University of Southern California, Los Angeles, LA, USA.

He is currently with the Vocational Training Council (VTC) of Hong Kong, Hong Kong, as the Deputy Executive Director. He is also the Adjunct Professor of the Chinese University of Hong Kong (CUHK), Hong Kong. Prior to joining VTC, he was a Professor and the Department Chairman of Mechanical and Automation Engineering of CUHK. His current research interests include computer vision and robotics.

Dr. Chung has served the academic communities in various capacities including the Chairman of the IEEE Hong Kong Section Joint Chapter on Robotics and Automation Society and Control Systems Society from 2001 to 2003. He is a fellow of HKIE, a fellow of BCS, CEng of the Engineering Council of U.K., and a member of MENSA.