High Precision Multi-touch Sensing on Surfaces using Overhead Cameras

Ankur Agarwal, Shahram Izadi, Manmohan Chandraker, Andrew Blake
Microsoft Research Cambridge, 7 J J Thomson Avenue, Cambridge, CB3 0FB

Abstract

We present a method to enable multi-touch interactions on an arbitrary flat surface using a pair of cameras mounted above the surface. Current systems in this domain mostly make use of special touch-sensitive hardware, require cameras to be mounted behind the display, or are based on infrared sensors used in various configurations. The very few that use ordinary cameras mounted overhead for touch detection fail to do so accurately due to the difficulty in computing the proximity of fingertips to the surface with a precision that would match the behaviour of a truly touch-sensitive surface. This paper describes a novel computer vision algorithm that can robustly identify finger tips and detect touch with a precision of a few millimetres above the surface. The algorithm relies on machine learning methods and a geometric finger model to achieve the required precision, and can be 'trained' to work in different physical settings. We provide a quantitative evaluation of the method and demonstrate its use for gesture based interactions with ordinary tablet displays, both in single user and remote collaboration scenarios.

Keywords: Interactive surfaces, bimanual interaction, multi-touch detection, hand gestures, computer vision

1. Introduction

Developing multi-touch technologies for interaction on tabletop surfaces is a very active area of research [1], the goal of which is to enable users to seamlessly interact with electronic media using finger touches and hand gestures. Several systems have been developed in this domain (*e.g.* [15,5,6,11]) and a large variety of configurations of sensing mechanisms and surfaces have been studied and experimented in this context. The most common of these include using specially designed surfaces with embedded sensors (*e.g.* using capacitive sensing [3,13]), cameras mounted behind a custom surface (*e.g.* [17]), cameras mounted in front of the surface (*e.g.* [9,10,18]) or on the surface peripheri (*e.g.* [14]). This paper addresses the case of overhead cameras mounted on top of a horizontal surface, using ordinary cameras that operate in the visible spectrum of light.

The overhead camera configuration has several advantages as it can be used to convert any arbitrary surface into an interactive one, thus allowing for smaller form-factor possibilities, easy installation and customization, and reduced costs. The use of ordinary cameras allows for various computer vision techniques to enable recognition of day-to-day objects or hand gestures [2] as well as to overlay physical objects from one workspace onto another in the case of remote collaboration setups, *e.g.* [8]. This can create a very rich and multipurpose workspace on the interactive surface.

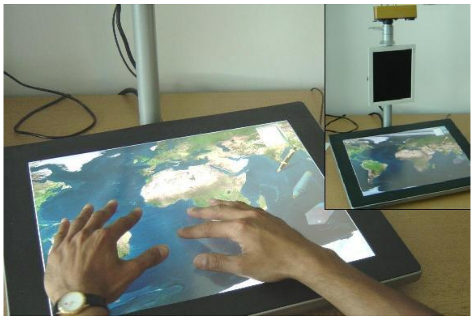

Figure 1. Our multi-touch sensing mechanism allows for enhanced gesture based interactions with an ordinary tablet display, simply by using an overhead stereo camera. The high precision is critical to giving the feel of a real touch-screen.

One of the major shortcomings of current overhead camera based systems, however, is the difficulty in accurately sensing contact with the surface. This limits the fluidity of interactions possible, *e.g.* the Visual Touchpad of [10] that uses two cameras for depth computation may report a touch event even if a finger is within approximately 1cm from the surface; in [9], the single camera system has no way to detect contact of a finger with the surface, so relies on detecting pauses in finger trajectories to report mouse *button* events. The PlayAnywhere system of [18] makes use of an infrared camera and a simple analysis of the shape of shadow of a finger to achieve good touch detection. This works well with projection based displays on opaque surfaces and when a finger is pointing in a direction almost perpedicular to that of the infrared light source, but we find that sensing multiple finger tips on top of an LCD display in the absence of directional lighting causes shadow based cues to be less reliable due to occlusions and lighting factors.

The contribution of this paper is a novel computer vision based algorithm that can robustly detect finger tips and sense touch for each finger with high precision using an overhead stereo camera. Unlike previous attempts to solve this problem [10,19], we present a quantitative analysis of both the finger detection and touch sensing components of our sys-

0-7695-3013-3/07 $25.00 © 2007 IEEE
DOI 10.1109/TABLETOP.2007.29

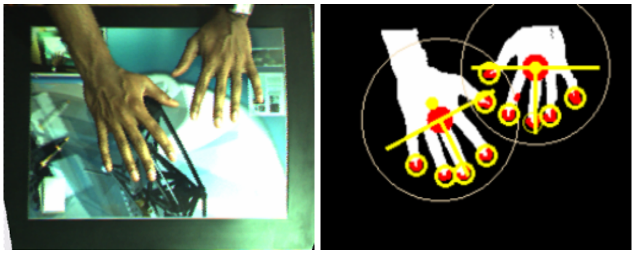

Figure 2. Fingertip detection and multi-touch sensing using stereo vision. *(left)* Input image from one camera after homography transformation; *(right)* Segmentation mask showing orientations of each hand and detected finger tips (bold circles indicate touch).

tem. We use the algorithm for enabling multi-touch interactions on tablet displays.

2. Physical Setup

Our system consists of a stereo camera mounted above a horizontal surface, viewing the surface at a slightly oblique angle to avoid interference within the working volume. The sensing algorithm can convert any surface into an interactive one, though this paper focusses on horizontal tablet displays to augment stylus input with multi-touch sensitivity. Figure 1 (inset) shows a prototype of the setup.

Calibration. In order to transform the camera view into the physical coordinates of the working surface, a corner detection algorithm is used to automatically detect the 4 corners of the working area in both the camera views. These may be the corners of the display screen as in our case, or pre-marked points on any surface. A homographic transformation and depth plane equation of the surface are then computed and stored for use during the main algorithm.

3. High Precision Touch Sensing

We develop a machine learning based approach to sense touching fingers on the surface. Labelled images of several different finger tips touching and not touching the surface are used as training data and a mathematical model is developed that *learns* to (a) detect multiple finger tips in an image, and (b) compute for each tip whether it touches the surface.

3.1. Image Segmentation

Before proceeding with fingertip detection, we first segment the hands from the rest of the image, which is referred to as the *background*. Other systems have used image differencing [9] or infrared filters [18] to suppress the background. Recent computer vision techniques that model the appearance statistics of the background [2] have proved very effective in dealing with arbitrary backgrounds. In this work, the background surface is an LCD screen. We exploit the fact that the light emitted by LCDs is polarized and make use of appropriately rotated polarizing filters on the stereo camera to cancel out the contents of the screen [7]. As a result, the display screen always appears to be ‘switched off’ to the cameras and the highly complex dynamic background can be suppressed by simple thresholding (figure 2).

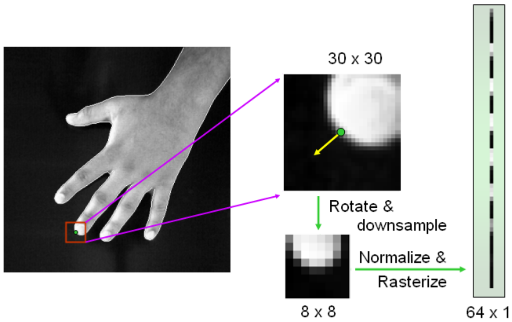

Figure 3. Encoding process that converts each point on the boundary of a hand into a signature vector. These are used to classify each edge-point as a *tip* or *non-tip* point, which are then spatially clustered to locate individual finger tips.

3.2. Fingertip Detection

Fingertip detection in the past has been done using shape filtering on binary images [9], finding strong peaks along hand blob perimeters [10] or using shadow based methods with heuristics that return a single fingertip detection per hand when the finger points in an appropriate direction [18]. In contrast, we make use of machine learning to develop a classifier that combines shape and appearance cues to robustly identify points having high probability of lying on a finger tip. These are called *tip points*, and are then clustered to obtain multiple fingertip detections in the image. Fingertips are thus detected only when there is a substantial evidence in the form of several tip points.

A few hundred points from the database of training images are marked as *tip* points or *non-tip* points (depending on whether or not they lie on a finger tip). These are encoded, using local image patches of 8×8 pixels, as 64-dimensional *signature* vectors. A linear decision rule in the form of a Support Vector Machine [16] is then learned that allows any new point to be classified as a tip point based on its signature. The signature computation process consists of normalizing each image patch with respect to rotation using the image gradient at that point, and scaling its intensity values to have unit variance. The matrix of intensities in this patch is then raster-scanned into a 64-dimensional vector. The process is illustrated in figure 3. This encoding allows the detection to be independent of the rotation of the finger and also quite robust to lighting variations. Individual finger tips are located and counted by performing a connected component analysis based clustering on the detected tip-points.

3.3. Touch and Hover Detection

Distinguishing events of touch from those where a finger is hovering a few millimeters above the screen requires very high precision stereo information. Conventional stereo algorithms that compute disparity images fail to provide this. In our physical setup, for instance, where the cameras have a baseline of 12cm and are mounted roughly 50cm above the tablet surface, depth near the screen is quantized every 5mm, so disparity images provide centimetre level precision at best. Here we develop an algorithm that probabilistically aggregates stereo cues from several points at each fingertip and uses a finger-specific model to achieve millimetre level precision.

Geometrically, the touching criterion is a function of the orientation and height of the finger tip above the screen. We extract this information by computing the equation of a plane that passes through points detected on the boundary of the finger – the plane that slices the finger to form its silhouette as shown in figure 4. However, in place of actual height above the screen, we use disparity values d_i relative to the screen surface for each pixel. At each point $\mathbf{x_i} \equiv (x_i, y_i)$ on the boundary of the finger tip (the *tip* points from above), the disparity is expressed as

$$d_i = \alpha^\top \mathbf{x}_i + \beta \tag{1}$$

where (α, β) are the parameters of the desired plane. $\alpha = [\alpha_1 \ \alpha_2]^\top$. $\mathbf{x_i}$ are measured in a local coordinate system attached to the finger, for rotation and translation invariance, and d_i is measured from stereo matching. With each disparity measure d_i, we also associate an uncertainity measure σ_i^2 which is obtained by modelling the stereo match likelihood [4] along each scan line as a normal distribution $\mathcal{N}(d_i, \sigma_i^2)$. This allows for a significant increase in the precision of estimated plane parameters (as compared to using *winner-take-all* stereo) since the optimal (α, β) may be estimated via a weighted least squares regression:

$$(\alpha^*, \beta^*) = \arg\min_{(\alpha,\beta)} \sum \frac{1}{\sigma_i^2} [d_i - (\alpha^\top \mathbf{x}_i + \beta)]^2 \tag{2}$$

In order to detect touch from the α and β values for each fingertip, we learn a linear decision rule on these parameters in the form of a discriminative classifier. The condition for touch thus takes the form

$$w_1\alpha_1 + w_2\alpha_2 + w_3\beta + w_4 > 0 \tag{3}$$

where $\{w_1 \dots w_4\}$ are weights that are learned using a second Support Vector Machine, taking labeled instances of touching and non-touching finger tips as training data. The linear form of a rule for detecting touch is motivated by geometrically approximating the finger tip as an ellipsoid that makes a rolling contact with the screen (see figure 4). In this case, the touching criterion may be expressed as the height of the centre of ellipsoid being less than a threshold:

$$[\alpha_1 \ \alpha_2] \begin{bmatrix} 0 \\ r \end{bmatrix} + \beta < r' \tag{4}$$

which is a special case of the condition in (3) with $w_1 = 0$. Learning the generic rule (3) directly from data rather than explicity modelling the geometry as in (4) allows for the model to accomodate deformability of the finger. Futhermore, it allows us to compute the probability that a fingertip touches the surface [12], and obtain more reliable information by incorporating temporal information.

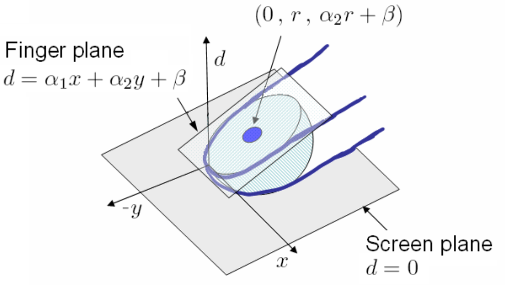
$(0, r, \alpha_2 r + \beta)$
Finger plane
$d = \alpha_1 x + \alpha_2 y + \beta$
d
$-y$
x
Screen plane
$d = 0$

Figure 4. Geometric model of a finger on the screen-plane. The orientation and height of a finger is summarized by a plane computed from the detected tip points; and the surface of the finger tip is well-approximated as an ellipsoid to detect instances of touch.

	Tips	Non-tips	Combined
average accuracy (%)	96.10	92.12	92.48
standard deviation	2.21	1.05	0.98

Figure 5. Classification accuracy of points as tip points or non-tip points without incorporating spatial information. More tip points are correctly classified than non-tip points. Although 92.48% represents a decent performance in itself, most mis-classifications are corrected during the spatial clustering step that follows (see text).

	Disparity histograms	Geometric finger model
average accuracy %	80.50	98.48
standard deviation	6.91	1.38

Figure 6. Classification accuracy of detected tips as touching or not touching the surface using two different stereo features.

4. Performance Evaluation

This section presents a quantitative evaluation of the fingertip detection and touch sensing accuracy of our system, followed by a qualitative description of its use for multi-touch and gesture-based interactions on tablet displays.

Sensing Accuracy. We conducted experiments with a database of 500 stereo images of a few different people's hands taken with the setup described in section 2. About 3 tip points were marked per visible finger on each image (each image had between 2 and 5 visible fingers) and 150 non-tip points marked per image. Figure 5 shows the classification accuracy of points as tip points or non-tip points in the form of an average over 100 trials with random 90%-10% splits into training and test data for cross-validation. The average accuracy of classifying points in this manner is 92.48%, but our clustering step removes almost all of

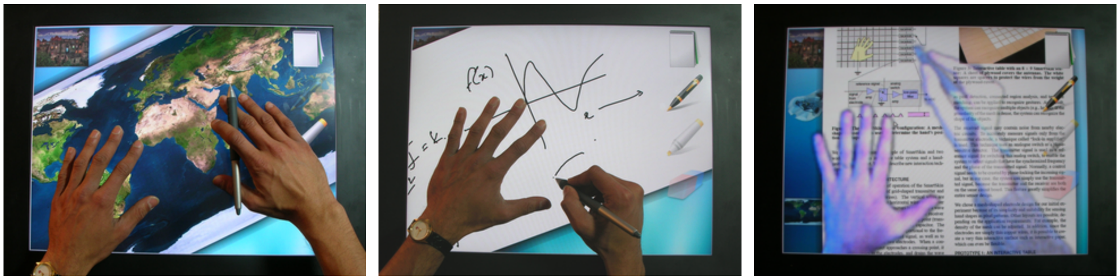

Figure 7. Our high precision algorithm allows seemless multi-touch interactions with an ordinary tablet display. *(left)* zooming into a map using a bimanual gesture *(centre)* using the non-dominant hand to rotate an electronic canvas while drawing with the stylus, and *(right)* a remote user zooms into a figure and highlights text in a collaborative work scenario where the workspace is shared multiple tablets. The stereo camera is used to render a remote user's hands on the tablet with depth-senstive transparency; we call this *phantom presence* [8].

the mis-classifications and individual finger tips are detected with almost 100% accuracy. For learning the touch-detection classifier, we define a finger tip that is roughly 2mm or more from the surface as non-touching. Figure 6 shows the accuracy of our touch-detection algorithm. Using the method described in section 3.3, detected fingertips were correctly classified as touching or not with an accuracy of 98.48%. As a benchmark, the table also shows the performance of a classifier than uses an alternate set of features based on simple disparity estimates from each fingertip as opposed to the geometric model of section 3.3.

Interaction on tablet displays. We implemented our algorithm to support bimanual interactions on tablet displays, as well as complement standard stylus input with gestures from the non-dominant hand. The sensing algorithm works at up to 20 fps on a 3.4GHz processor and our applications use interpolation to allow for seamless interactions. Figure 7 shows some of the interactions we currently support. The complete system is called the *Collaborative Slate* (C-Slate) and is described in [8]. Besides single-user interactions, it enables enhanced remote collaboration between multiple users on shared workspaces across more than one tablet; and supports object sensing and *phantom presence* [8].

5. Conclusion

This paper has presented a novel algorithm for multi-touch sensing on surfaces using an overhead stereo camera that supports multiple users. We have combined the use of machine learning with a geometrical intuition of the problem to robustly detect multiple finger tips in an image and sense touch with a precision of 2-3mm. This is a significant advancement over existing systems using stereo vision, which are restricted to centimetre level precision. The approach is also envisaged to be useful in other setups, *e.g*. using infrared images. The performance of the algorithm falls in adverse lighting conditions and is also currently susceptible to strong reflections on the screen surface. Although vision-based systems are often associated with such drawbacks, resolving these issues will be the focus of our future work.

[1] W. Buxton. Multi-Touch Systems I Have Known & Loved. 2007. http://www.billbuxton.com/multitouchOverview.html.

[2] T. Deselaers, A. Criminisi, J. Winn, and A. Agarwal. Incorporating On-demand Stereo for Real Time Recognition. In *Proc. Computer Vision and Pattern Recognition*, 2007.

[3] P. Dietz and D. Lehigh. DiamondTouch: a Multi-User Touch Technology. In *Proceedings of UIST*, 2001.

[4] V. Kolmogorov et al. Bi-layer segmentation of binocular stereo video. In *Proc. Computer Vision and Pattern Recognition*, 2005.

[5] J. Y. Han. Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection. In *Proceedings of UIST*, 2005.

[6] Tactex Controls Inc. http://www.tactex.com/products_array.php.

[7] H. Ishii, M. Kobayashi, and J. Grudin. Integration of Interpersonal Space and Shared Workspace: ClearBoard Design and Experiments. *ACM Trans. Information Systems*, 1993.

[8] S. Izadi, A. Agarwal, A. Criminisi, J. Winn, A. Blake, and A. Fitzgibbon. C-Slate: A Multi-Touch and Object Recognition System for Remote Collaboration using Horizontal Surfaces. In *IEEE TableTop Workshop*, 2007.

[9] J. Letessier and F. Berard. Visual Tracking of Bare Fingers for Interactive Surfaces. In *Proceedings of UIST*, 2004.

[10] Shahzad Malik and Joe Laszlo. Visual Touchpad: A Two-handed Gestural Input Device. In *Proc. ICMI*, 2004.

[11] Microsoft Surface Computing. http://www.surface.com.

[12] John Platt. Probabilities for Support Vector Machines. *Advances in Large Margin Classifiers*, pages 61–74, 1999.

[13] J. Rekimoto. SmartSkin: An infrastructure for freehand manipulation on interactive surfaces. In *Proc. CHI*, 2002.

[14] SMART Technologies. http://www.smarttech.com.

[15] TactaPad. http://www.tactiva.com/tactapad.html.

[16] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[17] A. Wilson. TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction. In *Proc. ICMI*, 2004.

[18] A. Wilson. PlayAnywhere: A Compact Interactive Tabletop Projection-Vision System. In *Proceedings of UIST*, 2005.

[19] C. Wren and Y. Ivanov. Volumetric Operations with Surface Margins. In *Computer Vision and Pattem Recognition Conference: Technical Sketches*, 2001.