HandSee: Enabling Full Hand Interaction on Smartphones with Front Camera-based Stereo Vision

Chun Yu[123],**Xiaoying Wei**[12], **Shubh Vachher**[1], **Yue Qin**[1], **Chen Liang**[1], **Yueting Weng**[1], **Yizheng Gu**[12], **Yuanchun Shi**[123]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Key Laboratory of Pervasive Computing, Ministry of Education, China
[3]Global Innovation eXchange Institute, Tsinghua University, Beijing, China
{chunyu, shiyc}@tsinghua.edu.cn,{wei-xy17,vachhers10, y-qin15,c-liang15,guyz17}@mails.tsinghua.edu.cn

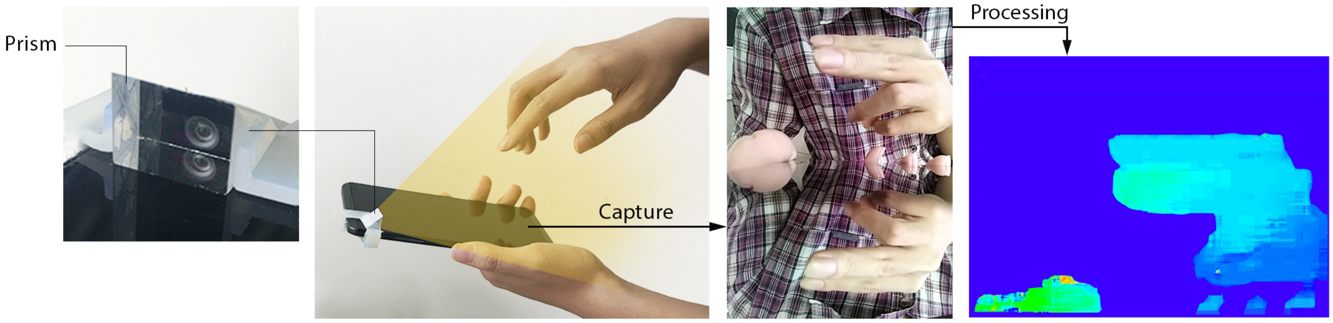

Figure 1: (a) The right angle prism mirror placed on the front camera. (b) The space above the touchscreen that can be covered. (c) A sample image captured by the front camera. (d) The derived depth image of the touching hand and the gripping hand.

ABSTRACT

We present HandSee, a novel sensing technique that can capture the state and movement of the user's hands touching or gripping a smartphone. We place a right angle prism mirror on the front camera to achieve a stereo vision of the scene above the touchscreen surface. We develop a pipeline to extract the depth image of hands from a monocular RGB image, which consists of three components: a stereo matching algorithm to estimate the pixel-wise depth of the scene, a CNN-based online calibration algorithm to detect hand skin, and a merging algorithm that outputs the depth image of the hands. Building on the output, a substantial set of valuable interaction information, such as fingers' 3D location, gripping posture, and finger identity can be recognized concurrently. Due to this unique sensing ability, HandSee enables a variety of novel interaction techniques and expands the design space for full hand interaction on smartphones.

CCS CONCEPTS

• **Human-centered computing** → **Smartphones**; **Touch screens**; **Gestural input**;

KEYWORDS

Smartphone interaction; full hand sensing; front camera; stereo vision; touching hand; gripping fingers

ACM Reference Format:
Chun Yu,Xiaoying Wei, Shubh Vachher, Yue Qin, Chen Liang, Yueting Weng, Yizheng Gu, Yuanchun Shi. 2019. HandSee: Enabling Full Hand Interaction on Smartphones with Front Camera-based Stereo Vision. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3290605.3300935

1 INTRODUCTION

Today, smartphone interaction is largely confined to the capacitive surface of the touchscreen. Numerous works have explored methods of overcoming this limitation ranging from enhancing the expressivity of touch (e.g. finger identification [19] and posture [21, 55]), to leveraging grip posture as an interaction context (e.g. interface shifting [8, 33]), as well as expanding the input space beyond the touchscreen surface

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CHI 2019, May 4–9, 2019, Glasgow, Scotland UK
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300935

(e.g. on the back [10, 54] or side [6, 8, 9] or around the device [7, 22, 24]). However, most of these works focus on interaction design. They either use dedicated sensing systems (e.g. Optitrack) or equip smartphones or users' hands with additional hardware sensors to provide specific and limited functionality.

In this paper, we present HandSee, a compact sensing technique that can capture rich information about hands and fingers interacting with a smartphone. We re-purpose the front camera by mounting a hypotenuse-coated right angle prism mirror on it, and direct it to look down along the screen's surface. As shown in Fig. 1, the field of view (FOV) covers the gripping fingers and the entire touching hand. The prism mirror provides two optical paths through which the front camera can look outward. This creates two virtual cameras that form a stereo vision system, as shown in Fig. 1.a. The stereo vision adds depth information that can further augment the sense ability.

To capture depth images of hands, we develop a pipeline of computer vision algorithms, which consists of four components: an efficient skin segmentation with online threshold calibration, stereo matching that reconstructs the depth image of scene over the touchscreen, and a merging algorithm that derives the depth image of hands. Based on the output, a set of valuable interaction information such as fingers' 3D location, gripping posture, finger identity can be derived, which enables a wide range of hand/finger interaction techniques on smartphones.

HandSee expands the space of full hand interaction on smartphones, which carries forward the idea that interprets users' intent beyond signals from the 2D capacitive screen [24]. We re-outline the interaction space into three subspaces: Touching Hand Only, Gripping Hand Only and Hand-to-Hand interaction. We propose a number of novel interaction techniques that fill in this space, and demonstrate the power of HandSee. Our user study shows that these techniques are well received by the users. They are easy to learn, convenient and fun to use.

Specifically, our contributions are threefold:

(1) A novel sensing scheme that captures both the touching hands and gripping fingers on a smartphone. We achieve stereo vision by placing a prism mirror on top of the front camera.
(2) A real-time pipeline to validate our setup's computational feasibility and compute the depth map of the user's hands, based on which, valuable interaction information can be derived.
(3) An expanded design space for full hand interaction on smartphones, as well as a number of novel interaction techniques.

In the remainder of this paper, we first review prior literature on hand/finger interaction and sensing. We then describe the hardware design of HandSee, followed by our algorithm pipeline. We move on to outline the design space and describe novel interaction techniques with feedback from a preliminary user study. We conclude this research with a discussion on the practicality, limitations and directions for future work.

2 RELATED WORK

In this section, we first review literature about enhancing hand/finger interaction on smartphones. Meanwhile, we discuss the sensing solutions in those works. We then give a brief introduction of general hand/finger sensing techniques, with a focus on camera-based ones.

Enhancing Hand/Finger Interaction on Smartphones

Expanding Expressivity of Finger Touch on Screen. A straightforward way to increase expressivity of touch is to leverage the state of touching finger. TapSense [21] recognizes the different parts of a human finger (e.g. tip, pad, nail and knuckle) tapping on the screen by analyzing sounds resulting from the tapping impact. Xiao et al. [55] describe a method that estimates the pitch and yaw of fingers relative to a touchscreen's surface based on the raw capacitive sensor data. DualKey [19] instruments the index finger with a motion sensor. It enables selection of letters on a miniature ambiguous software keyboard (e.g. a smartwatch) with different fingers.

In addition, a few works explored leveraging the above-screen space to improve interaction. Air+Touch [7] describes the concept of interweaving on-screen touch and in-air gestures to increase the expressivity of touch. The authors built a prototype system with a depth camera. Thumbs-Up [22] presents a similar idea that is specific to thumb input for one-handed interaction. Pre-touch [24] researches the potential of leveraging the status of the approaching finger, by increasing the sensing range of capacitive touchscreen. SegTouch [52] instruments the index finger with a touchpad, and allows users to perform thumb slides on it to define various touch purposes.

Interaction Beyond the Touchscreen. Researchers have explored extending smartphone interaction beyond the touchscreen surface. Some of these works enable input on the side or the back of the device. [35] detects finger taps on the sides of a smartphone using the built-in motion sensors. BackXPress [10] places a pressure sensitive layer on the back of the device, which allows pressure input on the back to augment the interaction with the remaining fingers on the front. Back-Mirror uses a mirror to reflect the back surface to the rear-facing camera of the phone, and recognizes hand gestures based on the visual pattern on the back surface.

Others explored the 3D space around the device. HoverFlow [28] uses infrared proximity sensors to track hands in the device's proximity. It can sense coarse movement-based gestures, as well as static position-based gestures. SideSight [4] embeds infra-red (IR) proximity sensors along the side of small device and supports single and multi-touch gestures in the space around the device. WatchSense [49] supports on and above-skin finger input for interaction on the move. The authors envisioned a depth sensor embedded in a wearable device to expand the input space. Song et al. [48] demonstrated how a computer vision algorithm can enable the rear-facing RGB camera to recognize in-air hand gestures along with a series of example applications.

Surround-See [57] presents a prototype that is similar to ours. It places an omni-vision lens on the front facing camera, and provides the smartphone with peripheral vision. The authors demonstrated various applications based on the ability to sense the environment and the objects of interest. These include detecting hand and recognizing hand gestures in the surrounding mid-air. In comparison, our work focuses on users' hands holding and touching the smartphone.

Grip Sensing and Grip-Aware Interaction. The grip posture of a user holding a smartphone (i.e., two-thumb, one-thumb and one-index) varies according to the form factor of the device size and interaction situations [12]. Meanwhile, grip posture is an important context for touch interaction on smartphones. It significantly affects users' input capability, such as the range of comfortable interaction and accuracy of touch, and so on [30]. To this end, a number of approaches have been proposed to sense grip posture. These works either use on-device sensors such as ones based on touch tracing [34], on-device motion sensor [41, 42] or motion sensor on the tapping hand [33] or a combination of the two [15, 18, 45], or by using additional capacitive sensors attached on the back of the device [6, 8, 9]. The basic principle is that different grip posture will result in corresponding holding or touch behavior that can be captured by these sensors. By taking advantage of classification algorithms such as a SVM, one can detect up to five different grip postures. Depending on the task, the detection precision, in current research, varies from 85%-99%.

Researchers have tried to incorporate grip posture as contextual information in order to enhance smartphone interaction. Grip-aware applications include improving the decoding algorithm of software keyboards [14], automatic interface orientation [8], interface shifting [33], endpoint prediction [41], automatically triggering applications [6], layout switching and continuous positioning [9].

Hand/Finger Tracking for General Purpose

Accurate hand/finger tracking is of great significance for human computer interaction. Various tracking techniques have been researched, such as using capacitive sensors [31], infrared signals [20], ultrasound [40], millimeter wave radar (i.e. Soli [32]), and monocular RGB camera [38] or depth camera [46] or a combination of the two. While some techniques can only detect motion (e.g. Soli and ultrasound), others can capture both motion and static posture.

Among these techniques, camera-based techniques show the greatest potential for providing complete scene information. In literature, there is a substantial body of work that deploys cameras on devices [7, 49], on the human body [5, 27, 36], and in the environment [46] to track hands. Commercially, LeapMotion, a stereo camera-based technique, has received wide attention in desktop and VR applications. However, no one has tried to use the available camera on smartphones to capture detailed information about the hands of the user interacting with the smartphone when they are positioned above the screen surface. We deem this would have great potential for practical use due to the wide use of current smartphone design.

Hand tracking is also an important and popular topic in computer vision research. Both model-based and machine learning-based approaches are researched. Model-based approaches rely on traditional model optimization and matching algorithms and require incorporation of extensive domain knowledge, such as hand kinematic models [51, 56], predicting spatial and temporal features for tracking [47], hand skeleton matching [11]. In contrast, machine learning based approaches do not need the researcher to define a model but need data from which they implicitly learn the required function. The limitation of machine learning approaches is the lack of high quality labelled data. Recently, researchers have tried to combine these two approaches. For example, GANerated Hands [37] uses a CNN to segment the hand from noisy background, and then applies model based optimization to estimate hand posture.

Compared with these works that intend to capture hands in arbitrary postures from arbitrary viewpoints, our problem space is much more limited as we focus on hand posture during interactions with a smartphone. The reduced problem space allows for better results. We use these previous research works as a strong platform as a proof of the technical feasibility of hand tracking.

3 THE OPTICAL DESIGN

In this section, we describe the optical model of HandSee, which explains how to achieve stereo vision of the scene above the touchscreen with a right angle prism mirror placed on the front camera.

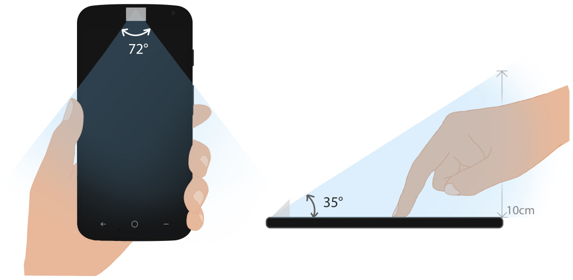

Figure 2: Field of view of the front camera and the interaction space above the touchscreen.

FoV of Camera

The right angle prism has a reflective coating on its hypotenuse. It mirrors the space above the touchscreen into the camera. To ensure that hands can be captured, the Field of Vision (FoV) of the camera should not be too small. For example, we calculate it to be at least 72 (landscape) x 35 (portrait) degrees for a 6-inch smartphone. The calculation is based on the assumption that the gripping hand is usually within the 10 cm range at the bottom of the screen. Therefore, as long as a user grips the phone in a normal way, the gripping posture can be accurately detected regardless of the gripping location. For those do not have such a wide camera, some gripping fingers might be missing, but the touching hand can usually be seen.

Front Camera-based Stereo Vision

In a typical stereo vision system, there are two cameras looking at the same scene. HandSee achieves this with two virtual cameras. As shown in Fig. 3, a single object has two optical paths into the cellphone camera, which projects onto different locations on the image plane. This equivalently creates two virtual cameras. Virtual Camera 1 is slightly above the touchscreen, which is the result of once-reflection on the hypotenuse of the prism mirror (the green line). Virtual Camera 2 is resulted from the light being twice reflected (the blue line), with the first reflection occurs on the touchscreen or the horizontal prism leg. The two virtual cameras, which look along the screen surface in parallel, form a stereo vision system.

The idea of providing stereo vision with a single camera is not new. Researchers have explored varying approaches using prisms or mirrors [13, 16, 25, 58]. However, in prior solutions, a prism was used equivalently as a mirror — either the two optical paths flying into the camera were reflected by two mirrors (or prisms) respectively, or one path flew directly into the camera and the other was reflected by a mirror (or a prism). In contrast, our approach uses a single prism to reflect two optical paths into the camera. The "total internal reflection" occurring on the inner side of the horizontal prism leg provides a high-quality image, which is of great importance for stereo estimation. In addition, to our knowledge, we are the first to apply this approach on a smartphone. In particular, we succeed in “rotating” the front camera for a screen parallel view and affording stereo vision at the same time with a single prism mirror.

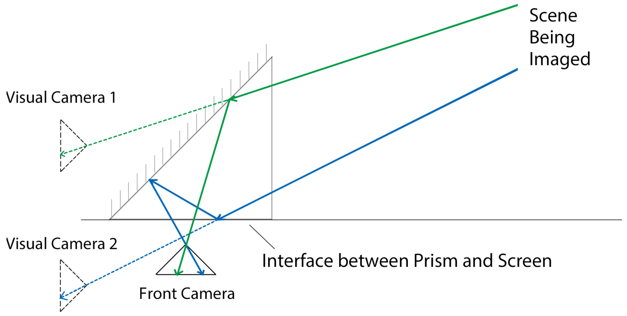

Figure 3: Front camera-based stereo vision: The green and blue lines each represent an optical path that light takes in travelling from the same object to the camera

Difference between Two Virtual Cameras

The image quality in the two virtual cameras are not equal. For light that goes into Virtual Camera 1 (the green line), total reflection occurs on the hypotenuse of the prism. The resulted image is of high quality, as if captured directly by the camera.

For Camera 2, the condition is more complex. The first reflection can occur either on the glass surface of the touchscreen or on the prism leg. For the former, the refractive index of the prism glass is higher than that of the air. This results in "total internal reflection" occurring on the inner side of the prism leg–the light cannot pass through and is entirely reflected. Thus, this part produces a high-quality image with equal brightness and sharpness to the image in Virtual Camera 1.

In contrast, light reflected on the touchscreen is attenuated due to partial reflection. This results in a relatively darker band in the captured image, as shown in Fig. 4. By adjusting the placement of the prism over the camera, we can minimize the height of the dark band. Later in this paper, we will describe how to remove the darkness with a brightness compensation function before performing stereo estimation.

4 THE ALGORITHMS OF HANDSEE

As a sensing technique, HandSee has two outputs: a depth image of hands and fingertip recognition results. Based on them, applications like gesture classification and full-hand interaction techniques can be further developed.

Fig. 5 illustrates the pipeline of deriving depth image of hands. Given a monocular RGB image, the pipeline first

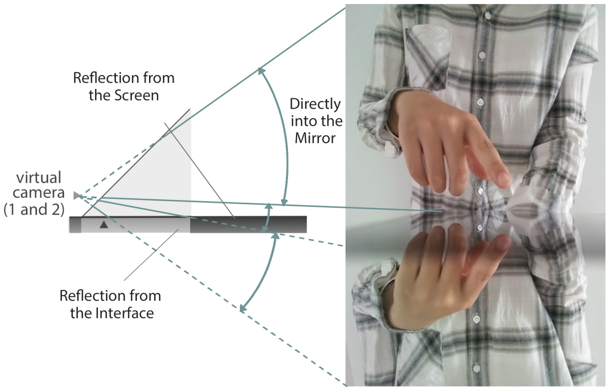

Figure 4: Different reflection rates on the prism leg and touchscreen of the smartphone, which leads to a darker band in the image of Virtual Camera 2

preprocesses it into two rectified stereo images. Then, color-based segmentation is applied to obtain a skin mask containing hands and fingers. Meanwhile, stereo estimation is performed to calculate a depth map for each pixel in the image. Next, the skin color mask and depth map are combined to produce a robust segmentation of the user's hand and fingers' areas in the input image. Note that the main purpose of the pipeline proposed in this paper is to validate the feasibility of online full hand sensing with the proposed sensing scheme. The algorithms should probably need to be refined or optimized if deployed on smartphones for practical use.

Pre-Processing

Since the cellphone surface does not reflect nearly as clearly as the mirror surface does, we see a significant difference in brightness levels between the images from the two views. To improve the accuracy of stereo estimation, we develop a simple color correction procedure for the image. We assume that the pixel color equation can be approximated as $Output = Input * R + L$ [2]. Here, R is the reflection factor, depending only on the physical properties of the phone surface and L is the surface luminescence of the phone. Since our camera is very close to the phone surface, the L term is nearly zero under normal lighting conditions. Using images collected against a white backdrop we used the least-squares estimation method to fit the parameter R for each pixel. We use this to remove the darker areas in each frame. Fig. 6 shows the result of removing the darker areas.

Note that although our data is collected when the screen is on, the impact of screen illumination is rather weak. That is because the virtual camera is low above the screen that the emergence angle is too large for illuminating effectively. This is why our pre-processing method works for different screen brightness.

Stereo Estimation of the Scene

We first use a checkerboard along with the OpenCV API [3] to estimate the focal length, optical center, distortion coefficient of the front camera and the rotation and translation relationship between the two cameras of the binocular vision system.

We chose an efficient, GPU based, implementation of the Semi-Global Matching algorithm [23] as our stereo matching algorithm because of its high computational speed and satisfactory accuracy. This algorithm outputs a depth estimation for each pixel using our binocular viewing system.

For our system, the theoretical estimation of depth error is $D^2/800$ (cm) [50], where D (cm) is actual depth, which means for a depth of 5-10cm, the error is 0.3-1.25 mm. Note that this estimation only represents a lower bound of depth error, supposing the calibration and stereo matching are done perfectly.

Skin Detection with Online Calibration

The stereo estimation is not perfect. For example, the background can falsely generate 3D points inside or near to the hand. Using skin color to filter such noise can increase the robustness of the result. The challenge is to account for the varying illumination conditions while restricting the hue and saturation ranges to eliminate background region as much as possible. Although skin detection has been extensively studied, the existing solutions ([43] for a review on color based pixel classification) do not satisfy our requirement due to their focus on recall of all possible human skin color as compared to our need for serving only one user at a time. Moreover, our skin detection modular requires high computational efficiency for real-time interaction.

Based on the above considerations, we design our skin detection modular with two components shown in Fig. 7: one to simply apply upper and lower hue and saturation thresholds to the image to segment the skin area, while the other one dynamically calibrates these upper and lower thresholds every few frames. In particular, we train a Convolutional Neural Network [1] to identify the skin pixels as a semantic segmentation task, making use of the FSD [43] and HGR [17, 26, 39] datasets.

Deriving Depth Image of Hands by Merging Skin Color Region and Depth Estimation

The focus of our skin segmentation algorithm is to maximize precision, i.e, minimize background region in segmentation. To then maximize recall, we combine the skin color mask with the depth estimation map to produce a robust segmentation mask of the hand and finger regions. First, we threshold the skin color mask by removing areas that have a depth value below an empirically obtained threshold value T (set

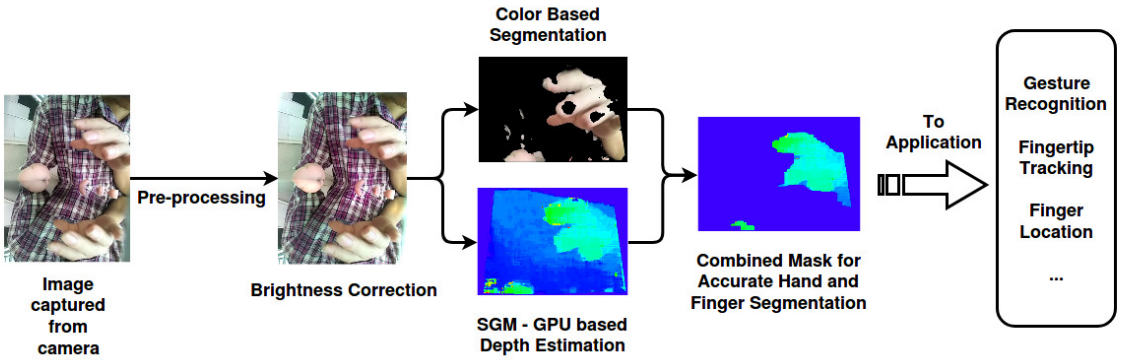

Figure 5: Pipeline of our real-time system for segmentation and stereo estimation of hands interacting with a smartphone

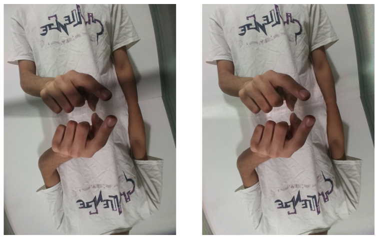

Figure 6: Compensating for difference in brightness

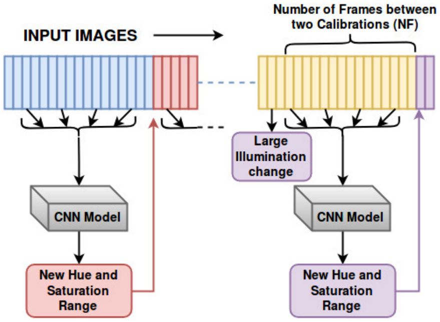
INPUT IMAGES
Number of Frames between two Calibrations (NF)
Large Illumination change
CNN Model
CNN Model
New Hue and Saturation Range
New Hue and Saturation Range

Figure 7: Online Skin Color Calibration updates color range periodically depending on user's skin color

as 22 in our experiments). This removes any remaining background regions that have similar colors to the user's skin.

We then use this result as a mask for our depth estimation output. We are left with the depth values of the user's hands and fingers. Using a Gaussian distribution centered around the mean of these depth values we discard pixels with depth values 1 (empirically obtained) standard deviation away from the mean on either side. This gives us a depth-based mask for the hand and finger regions excluding some high frequency noise pixels in the hand area.

The final result is a "bitwise or" operation of the depth-based mask and the skin color-minimum depth thresholded mask. We can use "bitwise or" safely because both our results are made using tight bounds to exclude regions other than the regions of interest. Qualitative analyses of many test cases have shown that our design produces a near perfect mask for hand and finger regions of the image.

Fingertip Recognition

Our fingertip recognition algorithm is inspired by the fingertip detection part of [7] and improved by incorporating more geometric features like shape, convex hull, moments and global maxima features as described in [29] and [44]. Compared with [7], in which the distance of hand points from the hand centroid directly is used for finding fingertips, our approach firstly figures out all the connected regions of the image and employs these extra features to find candidate regions of fingertips.

In specific, for every candidate region, we calculate the region centroid as well as the 3D distance from the region centroid of every contour point. Then we figure out all the distance maxima as fingertip candidates and filter out the false candidates using angle restrictions on the contour that the fingertip is a part to get a refined result. Incorporating feature matching increases the recall rate of detecting fingertips while spatial pattern restrictions and contour angle restrictions remove false positives, improving precision. The use of simple features and hierarchical feature matching also ensures the algorithm is also efficient in calculating results.

Evaluation

We put all the different pipeline sub-parts together calculate the frames-per-second (FPS) that our algorithm can process. The setup is tested on a server with 1 GTX 1080 Ti NVIDIA GPU with 12GB memory and 1 Intel i7-7700k CPU without taking network latency into account. The FPS for all parts of our pipeline is 30, specifically, 13 ms for pre-processing, 10ms for SGM-GPU, 4 ms for skin detection and deriving depth image of hand, 0.4ms for fingertip recognition, and 2.6ms for others.

For validating our setup's computational feasibility, our data collection included 10 users (5 males and 5 females) x 18

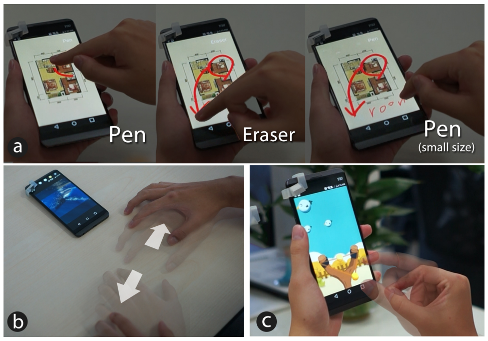

Figure 8: Touching Hand Only Interaction: (a) Painting Tool, (b) Table Touch, (c) Slingshot Game

tasks (screen tapping tasks with the 10 different fingers and 8 gripping postures) x 4 varied lighting scenarios (outdoor daylight, natural indoor light, bright focused lamplight, dim ambient lamplight) x 2250 frames (90 seconds for each sub task). We followed the leave-one-out cross-validation method to train and test classification models, the results are as follows: the accuracy of fingertip location is 98.0%, the accuracy of finger identification is 98.0% for 3 classes (left/right-thumbs and others), and 89.7% for 5 classes (left/right-thumbs, left/right-index and others); the accuracy of detecting gripping hand is 96.7%; the touch sensing range on the table surface is 20 cm below the bottom of the phone. Based on the output, a substantial set of valuable interaction information can be recognized simultaneously.

5 DESIGN SPACE AND EXAMPLE APPLICATIONS

HandSee provides three unique benefits when used to augment touch input. First, it covers a significant interaction space comprising of the space above the touchscreen. Second, it can sense the touching hand and the gripping fingers simultaneously. Third, it outputs the location of hands and fingers in 3D space. These benefits expand a promising design space of novel interaction techniques on smartphones. To better understand the potential, we divide the design space into three sub-spaces: Touching Hand Only, Gripping Hand Only, and Hand-to-Hand Interaction. Among these, the sub space of hand-to-hand interaction is novel, while the other applications are not new. We discuss the features and benefits of each sub-space and provide example applications. Our purpose in this section is not to enumerate all of them, but to illustrate the design possibilities and explore its potential.

Touching Hand Only

HandSee tracks the comprehensive state of the touching hand, including 3D location of the operating finger, finger identity and finger posture. Interaction techniques can leverage them for not only improving the input expressivity but also allowing applications to register inputs from beyond the touchscreen surface plane.

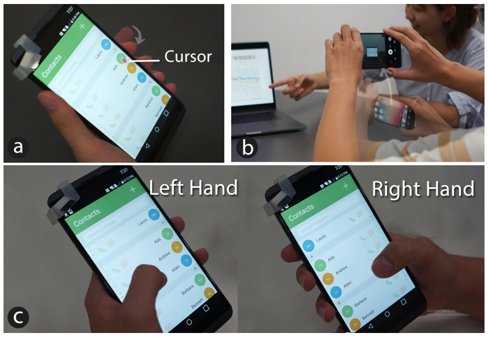

Figure 9: Gripping Hand Only Interaction: (a) Cursor Mode, (b) Camera, (c) Auto-UI

Finger Identification and Posture. Previous works have explored finger-worn sensors for finger identification [55] or estimating finger posture based on capacitive signals of the touchscreen [19]. HandSee supports these two concurrently and in a non-intrusive way. We've designed an example application that incorporates both these elements: a painting tool (Fig.8.a). The user can select different tools (e.g. pen, eraser and etc) with different fingers, and set the size of brush by adjusting the finger's angle against the touchscreen. This significantly reduces the time taken for selecting tools and parameters.

Extend Touch to Table Surface and Mid-Air. A touchscreen has a number of limitations such as the limited size, target occlusion during interaction [53], unresponsiveness to a wet hand and so on. When a phone is placed on a flat table, HandSee allows users to touch input on the table surface at the bottom of the phone (Fig.8.b). Users can perform tap, move, and multi-finger gestures (e.g. zoom and rotate) on this extra surface. Similarly, interaction can be extended into 3D, providing a more intuitive interaction with 3D objects. Fig.8.c shows a demo application: Slingshot Game. In the Game, a user pulls the slingshot, using 3D space to adjust the direction and intensity of releasing the bullet. This creates a more comfortable, immersive, and enjoyable experience.

Gripping Hand Only

The griping posture (i.e. which hand is used for holding the cellphone) is an important interaction context on smartphones. Previous research has investigated various sensing techniques to detect it (detailed in Section 2). In comparison, HandSee can not only identify gripping posture, but also track the location of the gripping fingers around a smartphone. This ability further enables novel input techniques.

Gripping Posture as a Context. Modern smartphones usually have a screen size that is not suitable for one-handed use.

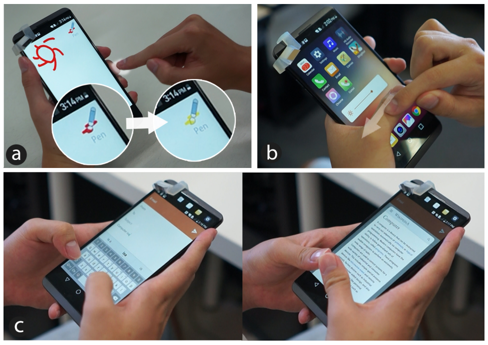

Figure 10: Hand-to-Hand Interaction: (a) FingerButton, (b) FingerBar, (c) Thumb-to-Thumb

One solution is to detect which hand is used, and adjust the UI layout to make needed widgets easier to acquire, as shown in Fig.9.c. Another potential is that we can define gripping posture-based shortcuts. For example, the gripping posture for "taking a photo" (Fig.9.c: two thumbs on one side of the phone, and other fingers on the opposite side) can automatically launch the camera. This can reduce the cumbersome and time-consuming process of finding the camera app on a smartphone to 1 second. Although these two concepts are not new, we deem them as representative and well reflecting HandSee' applicability.

Gripping Finger-based Gesture Input. When a user holds a phone in one hand, her gripping fingers still have the room and flexibility to move and she can, for example, move the index finger away from the screen edge. This creates the possibility for gripping finger-based gesture, an input space that has not been explored before. HandSee is well suited to this approach due to its ability to sense the movement of gripping fingers. To demonstrate the benefit, we propose a novel technique that allows users to switch between finger touch and cursor control by a stretching of the index finger. With this function, a user can easily switch to cursor mode and acquire an object that is out of thumb's reach, as shown in Fig.9.a.

Hand-To-Hand Interaction

An important feature of HandSee is that it can track the touching hand and the gripping fingers simultaneously. This opens the opportunity of hand-to-hand interaction on smartphones. For example, one hand can touch the other hand to perform inputs. In addition to adding expressivity, this approach can also reduce the participation of visual attention, by leveraging the proprioception between two hands. As a result, input can be easier and faster.

Finger Bar and Finger Button. A typical posture of interacting with a phone is one hand holding the phone and the other performing touch. In such a scenario, the gripping hand can be used as a touchable interface. To illustrate the concept, we propose two techniques. FingerBar (Fig.10.b) allows a user to slide a finger on the gripping thumb to provide input to a one-dimensional slider (e.g. controlling the volume). FingerButton allows a user to tap on gripping fingers as augmentations to on-screen buttons (Fig.10.a). Both techniques reduce required operational steps and augment the range of available input, thus increasing interaction efficiency.

Thumb-to-Thumb Gesture. Another typical phone interaction posture is bimanual: both hands hold the phone and perform touch input. For this scenario, we propose a Thumb-to-Thumb gesture, as an easy-to-perform and fast operation for mode switch or triggering a second view. Fig.10.c illustrates an example usage for enhancing typing experience. When filling a search query in a browser, a user might want to refer to a previous screen for a phone number or an address. Currently, the user has to switch back to the previous application, try to memorize the string and return to input. With thumb-to-thumb gesture, once two-thumb contact is detected, the system can put the previous screen on top of the browser so that the user can easily refer to the content. Then, he/she can release the two thumbs to input text. This provides a very efficient and lightweight way to toggle amongst modes on smartphones.

6 PROTOTYPE OF HANDSEE AND EXAMPLE APPLICATIONS

We prototype HandSee on an LG V20 smartphone (CPU: Quad-core 2.2 GHz, RAM: 4GB) running Android OS. The phone has a front camera on the upper left corner with an FoV of 100 x 80 degrees. The prism we used has dimensions 12mm x 10mm x 10mm. We 3D printed a housing to fix the prism on the smartphone. The total weight of the housing and prism is 15 g. Fig. 1 (a) shows the top view of our prototype, and Fig. 1 (c) shows an image captured when a user is interacting with our prototype.

Currently, we implement HandSee's algorithms on a PC server (1 GTX 1080 Ti NVIDIA GPU with 11GB memory) running Linux OS. On the smartphone, we develop an Android program running as a background service. It collects video from the front camera, and sends it, via WiFi, to the server in real time. The server processes the image and sends the recognition results back, which are further used by our customized applications to demonstrate our interaction techniques. To guarantee real-time performance, we downsize the image to 640x480. We tested the round trip latency from sending the image to receiving results on the phone to be 80ms (including 50ms delay of the network).

We prototype the nine interaction techniques described in Section 6, based on the depth image of hands and fingertips' 3D location. The purpose was to demonstrate the user experience of each technique. We first trained a CNN model to recognize 8 postures: left/right-hand gripping, one hand gripping and the other touching (2), two-hand gripping, thumb-to-thumb touch, single-hand grab (e.g. picking up a phone), and camera gripping posture. We measured the recognition accuracy to be 96.7%. Based on the posture result, we developed specific recognition/tracking algorithm for each application independently. For Auto-UI, Camera, and Thumb-to-Thumb, we directly use the posture result. For Painting Tool, the accuracy of identifying finger use (index vs. middle) was over 98%. For Table Touch and Slingshot Game, we directly read the 3D location of the fingertips. For FingerButton and FingerBar, we used the finger location and detected the contact of the touching finger and gripping fingers. For Cursor Mode, we analyzed the movement of the index finger (the one nearest to the camera).

7 INFORMAL STUDY

The goal of this study is to gain users' subjective feeback on the interaction techniques enabled by HandSee. We recruited 12 participants (5 females) aged between 20-28 years, from local campus to participate in the study. All participants used a touchscreen smartphone phone on daily basis and had owned one for more than 4 years.

We tested the nine example interaction techniques. Before the experiment, we introduced the working principle of HandSee. Then we tested the techniques one by one. For each technique, we first demonstrated our interaction technique. After that, a participant was allowed to use our technology freely. We finally interviewed him/her after each technique for their comments. In addition, participants were asked to indicate their agreement with four statements using a 7-point Likert scale. They are: 1) The input is convenient. 2) The interaction is easy to learn. 3) The technique is fun to use. 4) I want to have it on my phone.

Result

Fig.11 shows the subjective feedback of users regarding the four statements. Results showed that all nine techniques were well received by the participants. They were convenient to use (score=6.0), easy to learn (score=6.3), fun to use (score=6.2), and desired by the users (score=6.0). These results also indicated the room for interaction improvement on modern smartphones.

Users' comments also validate the design of our interaction techniques, and point to other applications and scenarios that can benefit from HandSee. We selectively report some of them.

Touching Hand interaction. Users appreciated using different fingers to choose different tools, since it saved them from repeated selection operations. One user suggested musical instrument applications (such as guitars): he can use different fingers to play different chords. Also, it is suggested that the tabletop can be used for playing two-player games or typing on a virtual keyboard. Users like to manipulate game objects in mid-air, as one user commented, "*it's feels really natural, the slingshot game is supposed to be played like that.*" Some users hoped to play AR games with mid-air hand gestures and believed it would be more immersive than before.

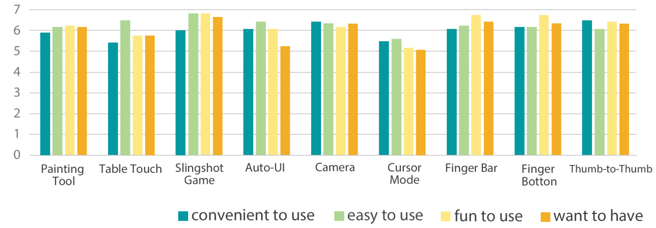

Figure 11: The subjective feedback of users. 1=Disagree strongly, 7=Agree strongly

Gripping Hand interaction. Users felt really convenient as the UI layout automatically adjusted after switching the gripping hand, making the common buttons appear in a position within reach. User thought that "*it allowed me to touch a comfortable position, and helped me in holding my phone more stably.*" They also appreciated that "*it doesn't require my active intervention, they're predictive!*". Besides, users felt comfortable to move the gripping finger, and said "*the finger is still flexible when holding a phone*". Users also expressed their willingness to associate other functions to their gripping fingers, such as shortcut for commonly used applications. Users particularly liked the idea of opening camera with a gripping gesture, thought that it could "*help them never miss a picture moment*", and could be very useful in many scenarios, such as shooting lively small animals, and the soon-to-be-switched PowerPoint.

Hand-to-Hand Interaction. Users felt that it was quite novel and reasonable to interaction with smartphone by touching on gripping hand. They commented "*touching any position of the thumb is easy with my other hand.*" They also reported that *"sliding on the fingers has tactile feedback, which is more comfortable than mid-air gestures."* Meanwhile, some users thought that touching on the finger button was "*as natural as taking paint with a brush while drawing.*" Users consistently liked using thumb-to-thumb gesture to evoke the thumbnails view of the previous application. Users thought this function was really practical, and the thumb-to-thumb gesture was very easy to learn, and did not interrupt the ongoing interaction.

8 DISCUSSION

In this section we discuss issues concerned with the deployment and adoption of HandSee in practice.

Form Factor

Overall, our current implementation of HandSee is compact. It only requires an additional right-angle prism mirror with a height of 10mm above front camera on the screen surface. According to our study, this prism does not disturb touch input on the major region of the touch screen and was accepted by all the participants. On the other hand, the height of prism can be further reduced. For example, if only the fingertips of the touching hand are of interest, the FoV in the vertical direction can be smaller. Besides, a convex mirror can also reduce the size of the prism but will distort the image and needs more dedicated calibration. Finally, it is best that a smartphone can be re-designed to contain the optical structure, for example, a motor-based mechanical structure to popup the prism when interaction is needed (e.g. vivo NEX S).

Hardware Support on Smartphone

HandSee imposes additional requirement for sensing and computing on a smartphone. But we think that it is consistent with development of technology: 1) *Sensing Hardware.* There emerges a trend to place additional cameras on commodity smartphones to provide better user experience (e.g., iPhoneX); 2) *Computing Software.* The state-of-the-art computer vision algorithms have demonstrated their capability in solving relevant problems (e.g., hand tracking via monocular RGB cameras [37]). In comparison, our problem is constrained on the space above a phone's touchscreen and might be less complex to solve. 3) *Computing Hardware.* Modern smartphones are incorporating hardware support for computer vision and machine learning algorithms that target real-time performance.

Power Consumption

Running computer vision algorithms on mobile devices always raises concerns of power consumption. While it might have been more controversial a decade ago, recently, more and more CV applications are being deployed on mobile devices, such as face recognition, AR/VR and so on. For HandSee, optimization of power consumption can be done by 1) using low-powered cameras (e.g. those with low resolution) and computing chips (e.g. hardware acceleration), and 2) turning on HandSee when needed, for example, in particular applications or only after a touch event is reported by the capacitive touch screen. This will reduce the computation load from continuous video processing to a single frame for per touch event.

RGB Camera versus Infrared Lighting and Sensing

We now implement HandSee using phones' front camera. The advantage is that the RGB image offers rich color information, which is important for our hand skin detection algorithm. However, the downside is that the algorithms heavily depends on illumination condition. Even if the camera screen can provide illumination, low illumination in the environment will still affect the quality of depth estimation.

An alternative solution is to adopt infrared lighting and sensing, which is widely used in practice (e.g., LeapMotion, Kinect and iPhoneX). Infrared lighting can guarantee the stability of illumination, and help filter out the background (by tuning the lighting power to illuminate hands and fingers in the near range, and leave the background in darkness). This would reduce the complexity of the recogntion algorithms. However, the shortcoming is that we lose the rich RGB information.

Privacy Issue

Camera use is a major source of privacy concern on smartphones. HandSee is no exception. Here, we analyze the risk. HandSee looks from the top down along the screen surface. In normal use, it would see users' chest as the background. Therefore, the privacy risk might be relatively lower compared with cases where a camera shoots at a wide open space. To further lower the risk, the aforementioned infrared solution might be preferred because the illumining distance can be controlled. Finally, we still think the best way to protect privacy is to guarantee that computation is done locally on smartphones.

9 LIMITATION AND FUTURE

The present research demonstrates the feasibility and potential of HandSee, but is still limited in the following aspects, which also point to possible avenues for future work.

First, our current implementation of HandSee algorithms is on PC, sending the recognition results back to the smartphone. For practical use, we need to research and deploy the algorithms on smartphones, with a special concern on the efficiency and power consumption.

Second, we need to further test and improve the performance of HandSee algorithms by taking various illumination condition and skin color into account. Also, the infrared lighting and sensing solution deserves exploration.

Third, the expanded space of full hand interaction needs further exploration. Future works can fill the space with more interaction techniques and conduct formal user studies to evaluate the usability of these techniques.

10 CONCLUSION

We present HandSee, a novel technique to sense and enable full hand interaction on smartphones. HandSee re-purposes the front camera to provide stereo vision, focusing on hands and fingers interacting with smartphone. HandSee is consistent with the recent hardware developments of smartphones, such as the increasing number of cameras, and on-device

GPU/NPU acceleration. We also contribute a reference implementation of the HandSee pipeline. The pipeline features online hand skin detection and stereo matching, which together provides robust sensing capabilities. The current implementation on PC can process 30 frames per second, providing real-time performance for interaction design. Future work is needed to deploy the algorithms on commodity smartphone devices. Thanks to the enhanced sensing capabilities, HandSee expands the design space of full hand interaction. We demonstrate a set of novel applications that are highly accepted by users. We believe HandSee will open new doors and enable more convenient and expressive interactions on smartphones.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Plan under Grant No. 2016YFB1001200, the Natural Science Foundation of China under Grant No. 61572276 and No. 61672314, Tsinghua University Research Funding No. 20151080408, and also by Beijing Key Lab of Networked Multimedia.

REFERENCES

[1] ahundt, aurora95, unixnme, and PavlosMelissinos. 2018. Keras-tensorflow implementation of Fully Convolutional Networks for Semantic Segmentation. https://github.com/aurora95/Keras-FCN.

[2] Tomas Akenine-Möller, Eric Haines, and Naty Hoffman. 2008. *Real-Time Rendering 3rd Edition*. A. K. Peters, Ltd., Natick, MA, USA. 1045 pages.

[3] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[4] Alex Butler, Shahram Izadi, and Steve Hodges. 2008. SideSight: Multi-"Touch" Interaction Around Small Devices. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, New York, NY, USA, 201–204. https://doi.org/10.1145/1449715.1449746

[5] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. Cyclopsring: Enabling whole-hand and context-aware interactions through a fisheye ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 549–556.

[6] Wook Chang, Kee Eung Kim, and Hyunjeong Lee. 2006. Recognition of Grip-Patterns by Using Capacitive Touch Sensors. 4 (2006), 2936–2941.

[7] Xiang 'Anthony' Chen, Julia Schwarz, Chris Harrison, Jennifer Mankoff, and Scott E. Hudson. 2014. Air+Touch: Interweaving Touch &#38; In-air Gestures. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 519–525. https://doi.org/10.1145/2642918.2647392

[8] Lung-Pan Cheng, Meng Han Lee, Che-Yang Wu, Fang-I Hsiao, Yen-Ting Liu, Hsiang-Sheng Liang, Yi-Ching Chiu, Ming-Sui Lee, and Mike Y. Chen. 2013. IrotateGrasp: Automatic Screen Rotation Based on Grasp of Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 3051–3054. https://doi.org/10.1145/2470654.2481424

[9] Lung-Pan Cheng, Hsiang-Sheng Liang, Che-Yang Wu, and Mike Y. Chen. 2013. iGrasp: Grasp-based Adaptive Keyboard for Mobile Devices. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13)*. ACM, New York, NY, USA, 2791–2792. https://doi.org/10.1145/2468356.2479514

[10] Christian Corsten, Bjoern Daehlmann, Simon Voelker, and Jan Borchers. 2017. BackXPress: Using Back-of-Device Finger Pressure to Augment Touchscreen Input on Smartphones. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4654–4666. https://doi.org/10.1145/3025453.3025565

[11] Martin de La Gorce, Nikos Paragios, and David J Fleet. 2008. Model-based hand tracking with texture, shading and self-occlusions. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference On*. IEEE, 1–8.

[12] Rachel Eardley, Anne Roudaut, Steve Gill, and Stephen J. Thompson. 2017. Understanding Grip Shifts: How Form Factors Impact Hand Movements on Mobile Phones. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4680–4691. https://doi.org/10.1145/3025453.3025835

[13] Joshua Gluckman and Shree K Nayar. 2002. Rectified catadioptric stereo sensors. *IEEE transactions on pattern analysis and machine intelligence* 24, 2 (2002), 224–236.

[14] Mayank Goel, Alex Jansen, Travis Mandel, Shwetak N Patel, and Jacob O Wobbrock. 2013. ContextType: using hand posture information to improve mobile touch screen text entry. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2795–2798.

[15] Mayank Goel, Jacob Wobbrock, and Shwetak Patel. 2012. GripSense: Using Built-in Sensors to Detect Hand Posture and Pressure on Commodity Mobile Phones. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 545–554. https://doi.org/10.1145/2380116.2380184

[16] Ardeshir Goshtasby and William A Gruver. 1993. Design of a single-lens stereo camera system. *Pattern Recognition* 26, 6 (1993), 923–937.

[17] Tomasz Grzejszczak, Michal Kawulok, and Adam Galuszka. 2016. Hand landmarks detection and localization in color images. *Multimedia Tools and Applications* 75, 23 (2016), 16363–16387. https://doi.org/10.1007/s11042-015-2934-5

[18] Hansong Guo, He Huang, Liusheng Huang, and Yue Sun. 2016. Recognizing the Operating Hand and the Hand-Changing Process for User Interface Adjustment on Smartphones. *Sensors* 16, 8 (2016).

[19] Aakar Gupta and Ravin Balakrishnan. 2016. DualKey: Miniature Screen Text Entry via Finger Identification. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 59–70. https://doi.org/10.1145/2858036.2858052

[20] Jefferson Y Han. 2005. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*. ACM, 115–118.

[21] Chris Harrison, Julia Schwarz, and Scott E. Hudson. 2011. TapSense: Enhancing Finger Interaction on Touch Surfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 627–636. https://doi.org/10.1145/2047196.2047279

[22] Khalad Hasan, Junhyeok Kim, David Ahlström, and Pourang Irani. 2016. Thumbs-Up: 3D Spatial Thumb-Reachable Space for One-Handed Thumb Interaction on Smartphones. In *Proceedings of the 2016 Symposium on Spatial User Interaction (SUI '16)*. ACM, New York, NY, USA, 103–106. https://doi.org/10.1145/2983310.2985755

[23] Daniel Hernandez-Juarez, Alejandro Chacón, Antonio Espinosa, David Vázquez, Juan Carlos Moure, and Antonio M. López. 2016. Embedded Real-time Stereo Estimation via Semi-Global Matching on the GPU. In *International Conference on Computational Science 2016, ICCS 2016, 6-8 June 2016, San Diego, California, USA*. 143–153. https://doi.org/10.1016/j.procs.2016.05.305

[24] Ken Hinckley, Seongkook Heo, Michel Pahud, Christian Holz, Hrvoje Benko, Abigail Sellen, Richard Banks, Kenton O'Hara, Gavin Smyth, and William Buxton. 2016. Pre-Touch Sensing for Mobile Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2869–2881. https://doi.org/10.1145/2858036.2858095

[25] Clemens Holzmann and Matthias Hochgatterer. 2012. Measuring distance with mobile phones using single-camera stereo vision. In *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*. IEEE, 88–93.

[26] Michal Kawulok, Jolanta Kawulok, Jakub Nalepa, and Bogdan Smolka. 2014. Self-adaptive algorithm for segmenting skin regions. *EURASIP Journal on Advances in Signal Processing* 2014, 170 (2014), 1–22. https://doi.org/10.1186/1687-6180-2014-170

[27] David Kim, Otmar Hilliges, Shahram Izadi, Alex D Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 167–176.

[28] Sven Kratz and Michael Rohs. 2009. HoverFlow: Expanding the Design Space of Around-device Interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '09)*. ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/1613858.1613864

[29] P Krejov and R Bowden. 2013. Multi-touchless: Real-time fingertip detection and tracking using geodesic maxima. In *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*. 1–7.

[30] Huy Viet Le, Sven Mayer, Patrick Bader, and Niels Henze. 2018. Fingers' Range and Comfortable Area for One-Handed Smartphone Interaction Beyond the Touchscreen. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 31.

[31] Huy Viet Le, Sven Mayer, and Niels Henze. 2018. InfiniTouch: Finger-Aware Interaction on Fully Touch Sensitive Smartphones. In *Proceedings of the 31th Annual ACM Symposium on User Interface Software and Technology (UIST'18). ACM, New York, NY, USA*, Vol. 13.

[32] Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. 2016. Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar. *ACM Trans. Graph.* 35, 4, Article 142 (July 2016), 19 pages. https://doi.org/10.1145/2897824.2925953

[33] Hyunchul Lim, Gwangseok An, Yoonkyong Cho, Kyogu Lee, and Bongwon Suh. 2016. WhichHand: Automatic Recognition of a Smartphone's Position in the Hand Using a Smartwatch. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16)*. ACM, New York, NY, USA, 675–681. https://doi.org/10.1145/2957265.2961857

[34] Markus Löchtefeld, Phillip Schardt, Antonio Krüger, and Sebastian Boring. 2015. Detecting Users Handedness for Ergonomic Adaptation of Mobile User Interfaces. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia (MUM '15)*. ACM, New York, NY, USA, 245–249. https://doi.org/10.1145/2836041.2836066

[35] William McGrath and Yang Li. 2014. Detecting Tapping Motion on the Side of Mobile Devices by Probabilistically Combining Hand Postures. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 215–219. https://doi.org/10.1145/2642918.2647363

[36] Pranav Mistry and Pattie Maes. 2009. SixthSense: a wearable gestural interface. In *ACM SIGGRAPH ASIA 2009 Sketches*. ACM, 11.

[37] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. Ganerated hands for real-time 3d hand tracking from monocular RGB. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 49–59.

[38] Sape Mullender (Ed.). 1993. *Distributed systems (2nd Ed.)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.

[39] Jakub Nalepa and Michal Kawulok. 2014. Fast and Accurate Hand Shape Classification. In *Beyond Databases, Architectures, and Structures*, Stanislaw Kozielski, Dariusz Mrozek, Pawel Kasprowski, Bozena Malysiak-Mrozek, and Daniel Kostrzewa (Eds.). Communications in Computer and Information Science, Vol. 424. Springer, 364–373. https://doi.org/10.1007/978-3-319-06932-6_35

[40] Suranga Nanayakkara, Roy Shilkrot, Kian Peen Yeo, and Pattie Maes. 2013. EyeRing: A Finger-worn Input Device for Seamless Interactions with Our Surroundings. In *Proceedings of the 4th Augmented Human International Conference (AH '13)*. ACM, New York, NY, USA, 13–20. https://doi.org/10.1145/2459236.2459240

[41] Matei Negulescu and Joanna McGrenere. 2015. Grip Change As an Information Side Channel for Mobile Touch Interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1519–1522. https://doi.org/10.1145/2702123.2702185

[42] Chanho Park and Takefumi Ogawa. 2015. A Study on Grasp Recognition Independent of Users' Situations Using Built-in Sensors of Smartphones. In *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15 Adjunct)*. ACM, New York, NY, USA, 69–70. https://doi.org/10.1145/2815585.2815722

[43] S. L. Phung, A. Bouzerdoum, and D. Chai. 2005. Skin segmentation using color pixel classification: analysis and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 1 (Jan 2005), 148–154. https://doi.org/10.1109/TPAMI.2005.17

[44] R. Meena Prakash, T. Deepa, T. Gunasundari, and N. Kasthuri. 2017. Gesture recognition and finger tip detection for human computer interaction. In *International Conference on Innovations in Information, Embedded and Communication Systems*. 1–4.

[45] Karsten Seipp and Kate Devlin. 2015. One-Touch Pose Detection on Touchscreen Smartphones. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*. ACM, New York, NY, USA, 51–54. https://doi.org/10.1145/2817721.2817739

[46] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Pushmeet Kohli, Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. 2015. Accurate, Robust, and Flexible Real-time Hand Tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3633–3642. https://doi.org/10.1145/2702123.2702179

[47] A. Sinha, C. Choi, and K. Ramani. 2016. DeepHand: Robust Hand Pose Estimation by Completing a Matrix Imputed with Deep Features. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 00. 4150–4158. https://doi.org/10.1109/CVPR.2016.450

[48] Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. 2014. In-air Gestures Around Unmodified Mobile Devices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 319–329. https://doi.org/10.1145/2642918.2647373

[49] Srinath Sridhar, Anders Markussen, Antti Oulasvirta, Christian Theobalt, and Sebastian Boring. 2017. WatchSense: On- and Above-Skin Input Sensing Through a Wearable Depth Sensor. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3891–3902. https://doi.org/10.1145/3025453.3026005

[50] George Stockman and Linda G. Shapiro. 2001. *Computer Vision* (1st ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.

[51] D. Tang, J. Taylor, P. Kohli, C. Keskin, T. Kim, and J. Shotton. 2015. Opening the Black Box: Hierarchical Sampling Optimization for Estimating Human Hand Pose. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 3325–3333. https://doi.org/10.1109/ICCV.2015.380

[52] Hsin-Ruey Tsai, Te-Yen Wu, Da-Yuan Huang, Min-Chieh Hsiu, Jui-Chun Hsiao, Yi-Ping Hung, Mike Y. Chen, and Bing-Yu Chen. 2017. SegTouch: Enhancing Touch Input While Providing Touch Gestures on Screens Using Thumb-To-Index-Finger Gestures. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. ACM, New York, NY, USA, 2164–2171. https://doi.org/10.1145/3027063.3053109

[53] Daniel Vogel and Patrick Baudisch. 2007. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 657–666.

[54] Pui Chung Wong, Hongbo Fu, and Kening Zhu. 2016. Back-Mirror: Back-of-device One-handed Interaction on Smartphones. In *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications (SA '16)*. ACM, New York, NY, USA, Article 10, 5 pages. https://doi.org/10.1145/2999508.2999522

[55] Robert Xiao, Julia Schwarz, and Chris Harrison. 2015. Estimating 3D Finger Angle on Commodity Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*. ACM, New York, NY, USA, 47–50. https://doi.org/10.1145/2817721.2817737

[56] Chi Xu, Lakshmi Narasimhan Govindarajan, Yu Zhang, and Li Cheng. 2016. Lie-X: Depth Image Based Articulated Object Pose Estimation, Tracking, and Action Recognition on Lie Groups. *CoRR* abs/1609.03773 (2016). arXiv:1609.03773 http://arxiv.org/abs/1609.03773

[57] Xing-Dong Yang, Khalad Hasan, Neil Bruce, and Pourang Irani. 2013. Surround-see: Enabling Peripheral Vision on Smartphones During Active Use. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 291–300. https://doi.org/10.1145/2501988.2502049

[58] Sooyeong Yi and Narendra Ahuja. 2006. An omnidirectional stereo vision system using a single camera. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, Vol. 4. IEEE, 861–865.